

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ

**КАФЕДРА СИСТЕМНОГО ПРОГРАМУВАННЯ І
СПЕЦІАЛІЗОВАНИХ КОМП'ЮТЕРНИХ СИСТЕМ**

«На правах рукопису»
УДК 004.93`1

«До захисту допущено»
Завідувач кафедри СПСКС

(підпис) В.П.Тарасенко
(ініціали, прізвище)
“ ” _____ 2018р.

Магістерська дисертація

на здобуття ступеня магістра

зі спеціальності 123 Комп'ютерна інженерія

(Комп'ютерні системи та компоненти)

на тему: «Система розпізнавання облич у відеопотоках на основі метода
Віюли-Джонса і локальних бінарних шаблонів»

Виконав: студент II курсу, групи _КВ-61м

(шифр групи)

Петричук Андрій Володимирович

(прізвище, ім'я, по батькові)

(підпис)

Науковий керівник

д.т.н., професор Зайцев В.Г.

(посада, науковий ступінь, вчене звання, прізвище та ініціали)

(підпис)

Рецензент _____

(посада, науковий ступінь, вчене звання, науковий ступінь, прізвище та ініціали)

(підпис)

Засвідчую, що у цій магістерській
дисертації немає запозичень з праць
інших авторів без відповідних
посилань.

Студент _____
(підпис)

Київ – 2018 року

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

Факультет прикладної математики
Кафедра системного програмування і спеціалізованих комп'ютерних
систем
Рівень вищої освіти – другий (магістерський)
Спеціальність 123 Комп'ютерна інженерія
(Комп'ютерні системи та компоненти)

ЗАТВЕРДЖУЮ
Завідувач кафедри СПСКС

В.П.Тарасенко
(підпис) (ініціали, прізвище)

«__» _____ 2018р.

ЗАВДАННЯ
на магістерську дисертацію студенту
Петричуку Андрію Володимировичу
(прізвище, ім'я, по батькові)

1. Тема дисертації «Система розпізнавання облич у відеопотоках на основі
метода Віюлі-Джонса і локальних бінарних шаблонів»,

науковий керівник дисертації **Зайцев В.Г., д.т.н., професор,**
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від «22» березня 2018 р. №986-с

2. Термін подання студентом дисертації 11 травня 2018 р.

3. Об'єкт дослідження - розпізнавання облич.

4. Предмет дослідження – алгоритми і методи виявлення облич в кадрі та
безпосереднього розпізнавання знайдених облич.

5. Перелік завдань, які потрібно розробити: аналіз існуючих методів
виявлення та розпізнавання облич; розпізнавання облич у відеопотоках,
дослідження ефективності різних варіацій локальних бінарних шаблонів
стосовно завдання розпізнавання облич в реальному часі, розробка
системи розпізнавання облич.

6. Перелік ілюстративного матеріалу: блок-схема реалізації розпізнавання облич методом найближчого сусіда, блок-схема алгоритму виводу інформації про облич на форму, UML діаграма класів, блок-схема алгоритму формування списків розпізнаних та нерозпізнаних облич, узагальнена блок-схема алгоритму обробки кадрів відеопотоку, блок-схема алгоритму обробки виявлених облич.

7. Перелік публікацій Основні положення і результати роботи будуть представлені на X науковій конференції магістрантів та аспірантів «Прикладна математика та комп'ютинг» ПМК-2018 (Київ, 21-23 березня 2018 р.) та I міжнародній науково-практичній конференції “Проблеми кібербезпеки інформаційно-телекомунікаційних систем” (PCSITS) (05-06 квітня 2018 р.).

8. Дата видачі завдання 5 вересня 2016 р.

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка
1	Огляд літератури по темі роботи	01.10.2016-05.12.2016	
2	Дослідження ефективності LBP операторів	05.02.2017-10.04.2017	
3	Розробка системи розпізнавання облич	10.08.2017-15.12.2017	
4	Тестування розробленої системи	20.12.2017-05.01.2018	
5	Оформлення пояснювальної записки	15.01.2018-10.04.2018	
6	Оформлення ілюстративного матеріалу	12.04.2018-24.04.2018	
	Попередній розгляд магістерської дисертації на кафедрі	26.04.2018	

Студент

(підпис)

(ініціали, прізвище)

Науковий керівник дисертації

(підпис)

(ініціали, прізвище)

РЕФЕРАТ

Актуальність теми. В останній час широке розповсюдження отримує відеоаналітика - технологія, яка використовує комп'ютерний зір для автоматизованого збору інформації по обличчям людей у послідовності кадрів, які отримуються з відеокамер в реальному часі чи з відеозаписів. Дана технологія може бути застосована в відеоспостереженні, системах безпеки, транспорті.

Рішення цієї задачі в першу чергу має безпосереднє використання в системах контролю доступу і ідентифікації особистості.

Об'єктом дослідження є процес розпізнавання облич.

Предметом дослідження алгоритми і методи виявлення облич в кадрі та безпосереднього розпізнавання знайдених облич.

Мета роботи: покращення розпізнавання облич в реальному часі. розпізнавання облич у відеопотоках, дослідження ефективності різних варіацій локальних бінарних шаблонів стосовно завдання розпізнавання облич в реальному часі, розробка системи розпізнавання облич.

Наукова новизна полягає в наступному:

1. Вперше запропоновано поєднання методу Віоли-Джонса для виявлення облич з методом на основі локальних бінарних шаблонів в якості алгоритма розпізнавання облич.
2. Проведено дослідження ефективності розпізнавання облич при використанні трьох варіантів LBP перетворень: класичний, рівномірний та центрально-симетричного.
3. Вперше доведено, що центрально-симетричний LBP оператор з розбиттям зображення на блоку по відношенню витрат пам'яті та ефективності розпізнавання являється 4x4.

Практична цінність розроблена система може використовуватись при вирішенні різних завдань відеоаналітики, і, в першу чергу, має безпосереднє застосування в системах контролю доступу і ідентифікації особистості.

Апробація роботи. Основні положення і результати роботи будуть представлені на X науковій конференції магістрантів та аспірантів «Прикладна математика та комп'ютинг» ПМК-2018 (Київ, 21-23 березня 2018 р.).

Структура та обсяг роботи. Магістерська дисертація складається з вступу, чотирьох розділів та висновків.

У вступі подано загальну характеристику роботи, зроблено оцінку сучасного стану проблеми, обґрунтовано актуальність напрямку досліджень, сформульовано мету і задачі досліджень.

У першому розділі розглянуто існуючі методи виявлення та розпізнавання облич, проаналізовано і поставлено задачу розпізнавання облич, описано функціональні вимоги до розроблюваної системи.

У другому розділі описуються основні етапи обробки кадрів відеопотоку системою, детально описані методи і алгоритми, які використовуються.

У третьому розділі зроблено порівняння ефективності різних варіацій локальних бінарних шаблонів стосовно задачі розпізнавання облич.

У четвертому розділі зроблено вибір інструментарію розробки, детально описується програмна реалізація системи, проводиться її тестування.

У висновках представлені результати проведеної роботи.

Робота представлена на 79 аркушах, 27 рисунків, містить посилання на список з 36 використаних літературних джерел.

Ключові слова: виявлення облич, розпізнавання облич, відеопотік.

РЕФЕРАТ

Актуальность темы. В последнее время широкое распространение получает видеоаналитика - технология, которая использует компьютерное зрение для автоматизированного сбора информации по лицам людей в последовательности кадров, получаемых с видеокамер в реальном времени или из видеозаписей. Данная технология может быть применена в видеонаблюдении, системах безопасности, транспорте.

Решение этой задачи в первую очередь имеет непосредственное использование в системах контроля доступа и идентификации личности.

Объектом исследования есть процесс распознавания лиц.

Предметом исследования алгоритмы и методы определения лиц в кадре и непосредственного распознавания найденных лиц.

Цель работы: улучшение распознавания лиц в реальном времени. распознавания лиц в видеопотоках, исследования эффективности различных вариаций локальных бинарных шаблонов относительно задачи распознавания лиц в реальном времени, разработка системы распознавания лиц.

Научная новизна заключается в следующем:

1. Впервые предложено сочетание метода Виолы-Джонса для определения лиц с методом на основе локальных бинарных шаблонов в качестве алгоритма распознавания лиц.
2. Проведено исследование эффективности распознавания лиц при использовании трех вариантов LBP преобразований: классического, равномерного и центрально-симметричного.
3. Впервые доказано, что центрально-симметричный LBP оператор с разбивкой изображения на блоки по отношению затрат памяти и эффективности распознавания является 4x4.

Практическая ценность. Разработанная система может использоваться при решениях различных задач видеоаналитики, и, в

первую очередь, непосредственное применение в системах контроля доступа и идентификации личности.

Апробация работы. Основные положения и результаты работы будут представлены на X научной конференции магистрантов и аспирантов «Прикладная математика и компьютеринг» ПМК-2018 (Киев, 21-23 марта 2018) и I международной научно-практической конференции "Проблемы кибербезопасности информационно-телекоммуникационных систем" (PCSITS) (05-06 апреля 2018).

Структура и объем работы. Магистерская диссертация состоит из введения, пяти глав и выводов.

Во *введении* представлена общая характеристика работы, произведена оценка современного состояния проблемы, обоснована актуальность направления исследований, сформулированы цели и задачи исследований.

В *первой* главе рассмотрены существующие методы обнаружения и распознавания лиц, проанализированы и поставлена задача распознавания лиц, описаны функциональные требования к разрабатываемой системе.

Во *второй* главе описываются основные этапы обработки кадров видеопотока системой, подробно описаны методы и алгоритмы, которые используются.

В *третьей* главе сделано сравнение эффективности различных вариаций локальных бинарных шаблонов относительно задачи распознавания лиц.

В *четвертой* главе сделан выбор инструментария разработки, подробно описывается программная реализация системы, проводится ее тестирование.

В *выводах* представлены результаты проведенной работы.

Работа представлена на 79 листах, 27 рисунков, содержит ссылки на список из 36 использованных литературных источников.

Ключевые слова: выделение лиц, распознавания лиц, видеопоток.

ABSTRACT

Actuality of theme. Recently, video analyst - technology that uses computer vision for the automated collection of information by the faces of people in a sequence of shots that are obtained from video cameras in real time or from videos - is widespread. This technology can be used in video surveillance, security systems, and transport.

The solution to this problem is, first and foremost, directly used in access control and identification systems.

The object of the study is facial recognition process.

The subject of the research are algorithms and methods for determining persons in the frame and directly recognizing the found persons.

The aim of the work: improve facial recognition in real time, facial recognition in video streams, the study of the effectiveness of various variations of local binary templates in relation to the face recognition problem in real time, the development of facial recognition system.

The practical value:

1. For the first time, a combination of Viola-Jones's method for revealing faces with a method based on local binary patterns as a face recognition algorithm is proposed.
2. The study of the effectiveness of face recognition using three variants of LBP transformations: classical, uniform, and centrally symmetric.
3. The efficiency of recognition with the use of a centrally symmetric LBP operator with respect to memory consumption and efficiency when splitting the image into 4x4 blocks was first established and demonstrated.

Approbation. The main provisions and results of the work will be presented at the Xth Annual Conference of Masters and Postgraduate Students "Applied Mathematics and Computer", PMK-2018 (Kyiv, March 21-23, 2018) and the 1st International Scientific and Practical Conference "Problems of Cybersecurity of Information and Telecommunication Systems "(PCSITS) (05-

06 April 2018).

The Structure and scope of work. The master's thesis consists of an introduction, four chapters and conclusions.

The introduction gives a general description of the work, makes an assessment of the current state of the problem, substantiates the relevance of the research direction, formulates the purpose and objectives of the research.

In the first section the existing methods of detection and recognition of faces are considered, the problem of face recognition is analyzed and put, functional requirements to the developed system are described.

The second section describes the main stages of processing the video stream by the system, describes in detail the methods and algorithms that are used.

In the third section, a comparison is made of the effectiveness of various variations of local binary patterns in relation to face recognition tasks.

In the fourth section the choice of development tools is made, the software implementation of the system is described in detail, its testing is carried out.

The conclusions are the results of the work.

The work is presented on 79 sheets, 27 drawings, contains a link to a list of 36 used literary sources.

Keywords: face detection, face detection, video streaming.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ	3
ВСТУП	4
1. ОГЛЯД МЕТОДІВ ВИЯВЛЕННЯ І РОЗПІЗНАВАННЯ ОБЛИЧ	8
1.1. Методи виявлення облич	8
1.2. Методи розпізнавання облич.....	19
1.2.1. Методи, засновані на значеннях яскравостей пікселів.....	19
1.2.2 Методи, засновані на характерних точках.....	24
1.3. Аналіз задачі розпізнавання облич у відеопотоках.....	26
2. ЕТАПИ ОБРОБКИ КАДРІВ ВІДЕОПОТОКУ	30
2.1. Виявлення облич методом Віюлі-Джонса	30
2.1.1. Інтегральне представлення зображень	30
2.1.2. Ознаки Хаара.....	31
2.1.3. Каскадна класифікація	33
2.2 Фільтр Гауса.....	34
2.3 LBP перетворення.....	36
2.3.1 Класичний алгоритм	36
2.3.2 Рівномірні LBP	38
2.3.3. Центральні-симетричні LBP.....	39
2.4 Маска значимих областей зображення.....	40
2.5 Метод найближчого сусіда.....	42
3. ДОСЛІДЖЕННЯ ЕФЕКТИВНОСТІ LBP ОПЕРАТОРІВ	45
3.1. Методика тестування	46
3.2. Результати дослідження.....	47
4. РАЗРОБКА СИСТЕМИ РОЗПІЗНАВАННЯ ОБЛИЧ	50
4.1. Інструментарій розробки	50
4.2. Аналіз потоків даних	51
4.3. Опис класів	52

4.3.1. RecognizedFace.....	54
4.3.2. LBPTransformer.....	55
4.3.3. FaceDetector.....	55
4.3.4. FaceClass.....	56
4.3.5. MainWindow.....	57
4.4. Опис інтерфейсу системи.....	65
4.5. Тестування розробленої системи.....	71
4.5.1. Тестування класифікатора облич.....	71
4.5.2. Порівняння швидкості роботи LBP операторів.....	72
ВИСНОВОК.....	75
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ.....	76
ДОДАТОК А. КОПІЇ ГРАФІЧНИХ МАТЕРІАЛІВ	
ДОДАТОК Б. КОПІЇ ПУБЛІКАЦІЙ ЗА ТЕМОЮ МАГІСТЕРСЬКОЇ ДИСЕРТАЦІЇ	
ДОДАТОК Г. ЛІСТИНГ ПРОГРАМИ	

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

AdaBoost (Adaptive Boosting) – це алгоритм підсилення класифікаторів шляхом об'єднання їх в «комітет».

SNoW (Sparse Network of Winnows) – алгоритм виявлення облич, який представляє собою двошарову мережу.

БНМ - багатошарова нейронна мережа.

НМ – нейронна мережа.

ЗНМ – згорткова нейронна мережа.

LBP (Local Binary Pattern) – локальні бінарні шаблони.

AAM (Active Appearance Models) – активні моделі зовнішнього вигляду.

ASM (Active Shape Models) – активні моделі форми.

XML (Extensible Markup Language) – розширювана мова розмітки.

UML (Unified Modeling Language) - уніфікована мова моделювання.

ВСТУП

На сьогоднішній день все більший інтерес визивають методи біометричної ідентифікації, які дозволяють визначити особистість людини по його фізичним характеристикам. Суттєвого поширення набули системи, які використовують для розпізнавання обличчя людини, пристрої які уже давно не є новими на ринку техніки, а саме за допомогою відеокамер.

В останній час широкого розповсюдження отримує відеоаналітика – технологія, яка використовує методи комп'ютерного зору для автоматизованого збору різної інформації на основі послідовності кадрів, які отримуються з відеокамер в реальному часі чи з відеозаписів. Дана технологія може застосовуватись у відеоспостереженні, системах безпеки, торгівлі, транспорті.

По оцінкам компанії MarketsandMarkets[1] в найближчі роки ринок відеоаналітики продовжить активно рости, а до 2020 складе 3971 мільйонів доларів. Даний напрямок активно розвивається.

Одною із задач, які вирішує відеоаналітика, є розпізнавання облич у відеопотоках. Рішення даної задачі в першу чергу має безпосереднє застосування в системах контролю доступу і ідентифікації особистості.

Одна з причин підвищеної уваги до біометричних технологій являється існування великої кількості соціальних і комерційних додатків, де можливі рішення названої проблеми будуть сприйняті досить успішно. Наприклад, зображення обличч отриманих з відеокамер зовнішнього спостереження, являється сьогодні важливим елементом документальної бази при розслідування злочинів, спостереження і розслідуванні кримінальних подій (автоматичне спостереження за підозрюваними, пошук людей, активне відеоспостереження при надзвичайних подіях і т. д.), а також в банківській сфері (банкоматах, системах віддаленого управління рахунком), ідентифікація людей по обличчю застосовується в системах контролю засвідченої особистості (паспортів, водійських прав,

імміграційних карт), інформаційній безпеці (доступ до ЕВМ і окремим програмам, базам даних, криптографічним додаткам, медичним відомостям, глобальної мережі Інтернет, системам електронної торгівлі).

Задача розпізнавання облич має серйозну практичну перспективу, так як цей метод ідентифікації особистості для людини є природнім і реалізовується на інтуїтивному рівні.

З точки зору обману системи сучасні методи ідентифікації по обличчю поки програють в надійності в порівнянні з ідентифікацією по райдужній оболонці ока, але рахується надійніших, ніж розпізнавання по відбиткам пальців чи геометрії зап'ястя.

Традиційні системи ідентифікації потребуються знання пароля, наявність ключа, ідентифікаційної карточки, чи іншого ідентифікаційного предмета, який можна забути чи згубити. На відміну від них, біометричні системи засновуються на унікальних біометричних характеристиках людини, які важко підробити і які однозначно визначають конкретну людину. До таких характеристик відносяться відбитки пальців, форма долоні, райдужна оболонка, зображення сітчатки ока, індивідуальні характеристики обличчя.

Незважаючи на безліч досліджень в цій області, які проводяться по всьому світу протягом останніх десятиліть, так і не було розроблено методів, що дозволяють надійно виявляти й розпізнавати обличчя людини за будь-яких умов.

Очікується, що ідентифікація людини по зображенню його обличчя буде найбільш використовуваною біометричною технологією. Оскільки для цього не потрібно спеціальне дороговартісне обладнання, не потрібний фізичний контакт з пристроями, не потрібно спеціально зупинятися і чекати опрацювання системи. Для певної якості роботи достатньо застосувати веб камеру. Звичайно одної веб-камери буде недостатньо, потрібна програма, яка оброблятиме отримані зображення та вирішуватиме згідно заданого алгоритму та методу роботи. Алгоритм роботи таких систем часто

повторяється, а ось методи зазвичай суттєво відрізняються. На даний час відомо та використовуються велика кількість методів та їх модифікацій. Поширеними є методи та їх похідні такі як метод Віоли-Джонса, еластичних графів, головних компонент, методи принцип яких базується на геометричному методі розпізнавання, методи засновані на нейронних мережах, такі як метод прихованої Маркової моделі, метод згорткової нейронної мережі, метод локальних бінарних шаблонів та ін.

Кожен із методів має свої переваги та недоліки, які проявляються в тих чи інших ситуаціях, що призводить до непередбачуваних випадків «пропустити чужого чи відмова у доступі для свого». Фактори, що впливають на якість роботи методів зазвичай одні і ті ж, та все ж таки кожен із методів має свою стійкість до певних із них. Такими факторами являється рівень та кут освітлення, відстань від камери, стан міміки чи здоров'я людини, настрій, кут нахилу обличчя, кут самого обличчя відносно камери, вікові зміни, наявність бороди чи вуса та ін.

Існуючі методи дозволяють вирішити ці задачі з неприйнятною затримкою у часі. Існують методи, за допомогою яких можна суттєво покращити час вирішення задачі, наприклад використання методу Віоли-Джонса в поєднанні з локальними бінарними шаблонами.

Дана робота присвячена розробці такої системи, яка вирішує цю задачу з використанням методу Віоли-Джонса і локальних бінарних шаблонів.

Оскільки об'єктом дослідження є конкретні методи розпізнавання образів, перша частина роботи представляє собою аналітичний огляд даних методів, а також порівняння з іншими відомими методами рішення поставленої задачі.

На наступному кроці проводиться аналіз задачі і формулювання вимог до розроблюваної системи. Дані вимоги дозволяються чітко визначити необхідний функціонал додатку, а також описати алгоритм його роботи.

Далі детально описуються основні етапи розроблюваної системи, в тому числі детектування облич в кадрі відеопотоку за допомогою метода Віюлі-Джонса і розпізнаванню знайдених облич методом найближчого сусіда з використанням локальних бінарних шаблонів в якості ознак класифікації.

Далі проводиться дослідження ефективності роботи різних модифікацій локальних бінарних шаблонів на наборі із 400 зображень облич. Порівняння отриманих результатів дозволяє виявити найбільш прийнятні набори ознак для класифікації. Зокрема, дане дослідження показало, що при значному зменшенні обчислювальних витрат, центрально симетричні локальні бінарні шаблони майже не поступаються в ефективності немодифікованим.

Після цього проводиться вибір інструментарію розробки і наводиться безпосередній опис програмної реалізації системи розпізнавання осіб з подальшим тестуванням продуктивності розробленої системи при різних настройках методів розпізнавання і наборах ознак.

1. ОГЛЯД МЕТОДІВ ВИЯВЛЕННЯ ТА РОЗПІЗНАВАННЯ ОБЛИЧ

Задачу розпізнавання облич в кадрах відеопотоку можна умовно розділити на два етапи. Перший етап – виявлення облич в кадрі. Другий етап – безпосередньо розпізнавання знайдених облич.

1.1. Методи виявлення облич

Виявлення обличчя означає побудову деякого набору правил, яким повинен відповідати фрагмент зображення, для того, щоб бути визнаним людським обличчям, наприклад: особа симетрична, риси обличчя (очі, ніс, рот) відрізняються від шкіри по яскравості, риси обличчя розташовані цілком певним чином. Також сюди можна віднести виявлення за допомогою шаблонів, заданих розробником. Виявлення обличчя за допомогою шаблону полягає в перевірці кожної з областей зображення на відповідність заданому шаблону.

З усього різноманіття існуючих алгоритмів виявлення облич можна виділити декілька актуальних методів[2], які заслуговують уваги. Розглянемо особливості, переваги і недоліки кожного з них.

Метод Віоли-Джонса був запропонований Полом Віолою і Майклом Джонсом в 2001 році[3] і став першим методом, який демонструє високі результати при обробці зображень в реальному часі.

Основні принципи, на яких заснований метод:

- Інтегральне представлення зображення. Це дає змогу швидко обчислювати необхідні об'єкти. Інтегральне представлення зображення - матриця, яка співпадає по розмірам з розмірами вхідного зображення в пікселях. Кожний елемент матриці представляє собою суму інтенсивностей всіх пікселів, які знаходяться лівіше і вище даного елемента.

- Використовуються ознаки, подібні до ознак Хаара[3], за допомогою яких виконується пошук потрібного об'єкта (в даному контексті, особливостей обличчя).

- Вибір ознак здійснюється з використанням алгоритму бустингу. Бустинг (від англ. – покращення, підсилення) – процедура послідовної побудови композиції алгоритмів машинного навчання, коли кожний наступний алгоритм намагається компенсувати недоліки композиції всіх попередніх алгоритмів.

- Класифікатор представляє собою функцію, яка визначає належність деякого досліджуваного об'єкта до певного класу. Класифікатор, який отримує на вхід ознаки, дає відповідь «істина» чи «брехня», дозволяє здійснювати виявлення.

- В процесі виявлення використовуються каскади ознак. В основі такої ідеї лежить побудова послідовності класифікаторів, яка називається каскадом: кожний наступний прагне врахувати помилки попереднього.

- Для зображень використовується метод на основі скануючого вікна: виконується сканування зображень вікном пошуку, після чого до кожного положення застосовується класифікатор.

Переваги:

- Алгоритм являється найпопулярнішим і широко розповсюдженим методом виявлення облич.

- Висока швидкість виявлення облич за рахунок використання каскадного класифікатора.

- Висока точність знаходження з дуже низьким відсотком помилкових спрацювань, порівняно з точністю виявлення у повільніших алгоритмах.

Недоліки:

- Необхідна велика навчальна вибірка і багато часу на навчання.

- Обмеження на положення обличчя при знаходженні.

AdaBoost (Adaptive Boosting) – це алгоритм підсилення класифікаторів шляхом об'єднання їх в «комітет». Вперше був запропонований Йоавом Фройндом і Робертом Шапіре в 1999 році[7]. Цей алгоритм може використовуватись в поєднанні з декількома алгоритмами класифікації для покращення їх ефективності. Даний алгоритм являється адаптивним в тому сенсі, що кожний наступний комітет класифікаторів будується по об'єктам, невірно класифікованим попередніми комітетами. Частіше всього використовується в поєднанні з іншими алгоритмами класифікації для їх підсилення (наприклад, як було описано раніше, в методі Віоли-Джонса).

Працює AdaBoost наступним чином. Кожну ознаку застосовується до кожного зображення. Відбираються ознаки з найменшою кількістю помилок.

Спочатку тестовим зображення присвоюється однакова вага, після кожної невірної класифікації вага зображень збільшується. Цей процес проходить до тих пір, поки не досягнута потрібна точність, або досягнутий певний коефіцієнт помилок, або відібрана певна кількість ознак.

Після даної операції кількість ознак значно скоротилась, але в фотографіях, в багатьох випадках, обличчя займає маленьку область, і щоб не запускати всі ознаки, які не потрібні для даної області, ввели поняття каскад класифікатор. Тепер ознаки групуються по стадіям. Перші стадії містять малу кількість ознак, в кожній наступній стадії їх кількість буде збільшуватись. Область, яка пройшла всі стадії, являється обличчям.

Переваги:

- Алгоритм підлаштовується під проблемні елементи навчальної вибірки. Теоретично метод досягає нульової помилки навчання за кінцеве число ітерацій.
- Висока швидкість роботи.
- Простота реалізацією.

Недоліки:

- Алгоритм чутливий до шумів і викидам даних.
- Потрібно багато часу на навчання, яке залежить від кількості класифікаторів і розміру навчальної вибірки.

SNoW (Sparse Network of Winnows)[5] – алгоритм виявлення облич, який представляє собою двошарову мережу, вхідний шар якої складається з вузлів, кожний з яких відповідає деякій характеристиці вхідного зображення, вихідний в свою чергу складається всього з двох вузлів, кожний з яких відповідає розпізнавальним класам зображення (генерує 1 при наявності деякої особливості та 0 у разі її відсутності на зображення), вихідний же складається всього з двох вузлів, кожен з яких відповідає розпізнаваним класам зображень («обличчя», «не обличчя»)[5]. В якості ознак в даному алгоритмі використовуються SMQT (Successive Mean Quantization Transform) ознаки, прапори рівності певним величинам середнього значення і дисперсії в кожному з прямокутних фрагментів зображення розміром 1x1, 2x2, 4x4 і 10x10 (усі зображення мають розмір 20x20 пікселів). Це дає простів ознак розмірності 135424. При проведенні класифікації на вхідні вузли подається інформація про присутність певних характеристик в оброблюваному зображенні. Вузли вихідного шару обчислюють лінійну комбінацію сигналів, які генеруються вхідними вузлами. Коефіцієнти лінійної комбінації задаються вагами між вхідними вузлами. Дані перетворення дозволяють вилучити з локальної області зображення складову, яка не залежить від освітленості. Воно полягає в квантуванні області зображення з порогом квантування рівним середньому значенню пікселів, які входять в цю область[6].

SNoW спеціально розроблено для випадків класифікації, коли потенційне число характеристик об'єктів, важливих для класифікації може бути дуже велике, але невідомо заздалегідь. Розріджена архітектура мережі дозволяє використовувати величезну кількість властивостей зображення в якості вхідних даних, оскільки в процесі тренування всі несуттєві

характеристики відкидаються, і не уповільнюють зрештою, функціонування класифікатора[30].

При перевищенні заданого порогу, приймається рішення про наявність обличчя на зображенні. Переваги:

- Використовувані ознаки не чутливі до змін освітленості в локальних областях зображення.
- Висока швидкість за рахунок просіювання компонент вектору ознак.
- Висока точність виявлення.

Недоліки:

- Алгоритм чутливий до шумів і викидам даних.

Нейромережеві методи включають в себе цілий клас алгоритмів. Основна ідея – послідовність перетворень сигналу паралельно працюючими функціональними елементами, нейронами. Сутність процесу навчання таких мереж зводиться до зменшення середньоквадратичної помилки. Системи виявлення об'єктів на зображенні, засновані на нейронних мережах, використовують ієрархічну структуру. Спочатку вектор ознак оброблюється грубою мережею з високим рівнем помилок другого роду, далі, якщо вектор не був класифікованим як не об'єкт, рішення корегується більш точною та повільнішою мережею[7].

Переваги:

- Висока точність виявлення при правильному налаштуванні параметрів мережі.

Недоліки:

- Чутливість до шуму.
- Необхідність в ретельному налаштуванні параметрів нейронної мережі для отримання хороших результатів.
- Схильність до повторного навчання.

- Висока обчислювальна складність, і, як результат, швидкість роботи, недостатня для обробки в реальному часі.
 - Досить тривала процедура навчання.
- Багатошарові нейронні мережі.

Архітектура багатошарової нейронної мережі (БНМ) складається з послідовно з'єднаних шарів, де нейрон кожного шару своїми входами пов'язаний з усіма нейронами попереднього шару, а виходами – наступного. Нейронна мережа з двома вирішальними шарами може з будь якою точністю апроксимувати будь яку багатовимірну функцію.

Нейронна мережа (НМ) з одним вирішальним шаром здатна формувати лінійні розділяючі поверхні, що сильно звужують коло завдань, які вони вирішують, зокрема, така мережа не зможе вирішити завдання типу "виключає або". Нейронна мережа з нелінійної функцією активації і двома вирішальними шарами дозволяє формувати будь-які опуклі області в просторі рішень, а з трьома вирішальними шарами - області будь-якої складності, в тому числі і неопуклої. При цьому багатошарова нейронна мережа не втрачає своєї узагальнюючої здатності. Навчаються БНМ за допомогою алгоритму зворотного поширення помилки, що є методом градієнтного спуску в просторі ваг з метою мінімізації сумарної помилки мережі. При цьому помилки (точніше, величини корекції ваг) поширюються в зворотному напрямку від входів до виходів, крізь ваги, що з'єднують нейрони. Найпростіше застосування одношарової НМ (так званою автоасоціативною пам'яттю) полягає в навчанні мережі відновлюванні зображень, які подаються на вхід. Подаючи на вхід тестове зображення і обчислюючи якість реконструйованого зображення, можна оцінити, наскільки мережа розпізнала вхідне зображення. Позитивні властивості цього методу полягають в тому, що мережа може відновлювати спотворені і зашумлені зображення, але для більш серйозних цілей він не підходить. БНМ також використовується для безпосередньої класифікації зображень - на вхід подається або саме зображення в будь-якому вигляді, або набір

раніше витягнутих ключових характеристик зображення, на виході нейрон з максимальною активністю вказує приналежність до розпізнаного класу.

Нейрон з максимальною активністю вказує приналежність до розпізнаного класу. Якщо ця активність нижче деякого порогу, то рахується, що наданий образ не належить до одному з відомих класів. Процес навчання встановлює відповідність наданих на вхід образів з приналежністю до вказаного класу. Це називається «навчання з учителем». У застосування до розпізнавання людини по зображенню обличчя таких підхід добре застосовувати для задачі контролю доступу невеликої групи облич. Він забезпечує безпосереднє порівняння мережею самих обрізав, але зі збільшенням класів час навчання і роботи мережі збільшується експоненціально, і тому, для таких задач, як пошук людини у великій базі даних потребує виявлення компактного набору ключових характеристик, на основі яких можна проводити пошук.

В частості, БНМ можуть використовуватись для класифікації облич на основі таких характеристик, як відстань між деякими специфічними частинами обличчя(ніс, рот, очі). Також використовуються гібридні системи, наприклад об'єднання з марківською моделлю. В класичній БНМ міжшарові нейронні об'єднання повнозв'язні, і зображення представлене у вигляді одновимірного вектору, хоча воно двовимірне.

Архітектура згорткової нейронної мережі (ЗНМ) направлена на подолання цих недостатків. У ній використовується локальні рецепторні поля (забезпечує локальну двовимірну зв'язність нейронів), загальні ваги (забезпечують детектування деяких рис в будь якому місці зображення) і ієрархічна організація з просторовими підвибірками.

Згорткові нейронні мережі забезпечують часткову стійкість до змін масштабу, зміщення, поворотів, спотворення. Архітектура ЗНМ складається з багатьох шарів, кожний з яких має декілька площин, причому нейрони наступного шару з'язані лише з невеликою кількістю нейронів попереднього шару з околиці локальної області. Ваги в кожній точці одної

площини однакові (згорткові шари). За згортковим шаром слідує шар, який зменшує його розмірність шляхом локального усереднення. Потім знову згортковий шар, і так далі. Таким чином досягається ієрархічна організація. Останні шари витягують найбільш загальні властивості, менше залежать від спотворень зображення. Навчаються ЗНМ стандартним методом оберненого розповсюдження помилки.

У порівнянні з БНМ, ЗНМ має істотні переваги як по швидкості, так і по надійності класифікації. Корисною властивістю ЗНМ є те, що характеристики, які формуються на виходах верхніх шарів ієрархії, можуть бути застосовані для класифікації найближчого сусіда, причому ЗНМ може успішно вилучати такі характеристики і для образів, які відсутні в навчальному наборі.

Нейронна мережа Хопфілда.

Алгоритм навчання мережі Хопфілда істотно відрізняється від класичних алгоритмів навчання персептронів тим, що замість послідовного наближення до потрібного стану з обчисленням помилок, всі коефіцієнти вагової матриці розраховуються по одній формулі, за один цикл, після чого мережа одразу готова до роботи. НМ Хопфілда являється одношаровою і повнозв'язною (зв'язки нейронів на самих себе відсутні), її виходи зв'язані з входами.

У відмінності від БНМ, НМ Хопфілда є релаксаційною – тобто будучи встановленою в деяке початкове положення функціонує до тих пір, поки не досягне стабільного стану, яке і буде являтися її вихідним значенням.

Обмеження методу:

- образи, які запам'ятовуються не повинні бути дуже схожими один на одного;
- зображення не має бути зміщене або повернуте відносно його вихідного стану.

Для подолання цих недолатків розглядаються різні модифікації класичної нейронної мережі Хопфілда. Мережа Хопфілда з ортогональним

перетворенням дозволяє відновити сильно скорельовані образи за рахунок перетворення в їх вихідну множину до дуальної множини векторів. Таким чином, виходить нейронна мережа, яка може запам'ятовувати деяку кількість векторів, і при подачі на вхід будь якого вектору, можна визначити, на який із запам'ятованих він схожий найбільше.

Прихована марківська модель — статистична модель, що імітує роботу процесу, схожого на марківський процес з невідомими параметрами. Згідно моделі ставиться задача знаходження невідомих параметрів на основі інших параметрів, за якими ведеться спостереження. Отримані параметри можуть бути використані в подальшому аналізі для розпізнавання облич.

З точки зору розпізнавання – зображення це двомірний дискретний сигнал. Важливу роль в побудові моделі зображення грає вектор спостереження. Для того, щоб уникнути розбіжностей в описах, зазвичай використовують прямокутне вікно для розпізнавання. Щоб не втрачати області даних, прямокутні вікна мають перекривати одне інше. Значення для перекривання, як і області розпізнавання підбираються експериментально. Після зняття блоку виконують його перетворення в цифровий блок за одним з двох методів [33]:

- Карунена-Лоева (КЛП);
- дискретне косинусне перетворення (ДКП).

Support vector machine[8] - метод опорних векторів є машинним алгоритмом, який навчається на прикладах та використовується для класифікації об'єктів.

В основі методу опорних векторів лежить математична сутність – алгоритм максимізації деякої математичної функції відносно наявного набору даних.

Метод застосовується для зниження розмірності простору ознак, не приводячи до істотної втрати інформативності тренувального набору об'єктів. Застосування методу головних компонент до набору векторів лінійного простору, дозволяє перейти до такого базису простору, що

основна дисперсія набору буде спрямована уздовж декількох перших осей базису, званих головними осями. Натягнутий на отримані таким чином головні осі підпростір є оптимальним серед усіх просторів у тому сенсі, що найкращим чином змальовує тренувальний набір. Це набір алгоритмів схожих з алгоритмами виду «навчання з учителем», що використовується для задач класифікації та регресійного аналізу. Цей метод належить до сімейства класифікаторів.

Метод опорних векторів заснований на тому, що шукається лінійний поділ класів. Мета тренування більшості класифікаторів – мінімізувати помилку класифікації на тренувальному наборі (звану емпіричним ризиком). На відміну від них, за допомогою методу опорних векторів можна побудувати класифікатор, мінімізуючий верхню оцінку очікуваної помилки класифікації (у тому числі і для невідомих об'єктів, що не входили в тренувальний набір).

Застосування методу опорних векторів до задачі виявлення особи полягає у пошуку гіперплощин в просторі ознак, що відокремлює клас зображень облич від зображень «не облич». Можливість лінійного поділу таких складних класів, як зображення облич та «не облич» є малоімовірною. Одна класифікація за допомогою опорних векторів дозволяє використовувати апарат ядерних функцій для неявного проектування векторів-ознак у простір потенційно більш високої розмірності (ще вище, ніж простір зображень), в якому класи можуть виявитися лінійно розподіленими. Неявне проектування за допомогою ядерних функцій не призводить до ускладнення обчислень, що дозволяє успішно використовувати лінійний класифікатор для лінійно нероздільних класів[31,32].

У результаті роботи системи виділення осіб можуть виникати два типи помилок: не виділення обличчя та хибне виявлення (виділення об'єкта, який обличчям не являється). У зв'язку з наявністю двох типів помилок, існує два основних параметри, що характеризують ефективність роботи

алгоритмів виявлення обличчя: рівень виявлення, що показує відсоток виявлених облич, і рівень невірного виділення, рівний загальному числу помилкових виявлень на всьому тестовому наборі.

Суть методу полягає в знаходженні гіперплощини, що розділяє два класи. При цьому з усіх можливих гіперплощин, які поділяють два класи, необхідно вибрати таку гіперплощину, відстань до якої від кожного класу була максимальною. Ця гіперплощина називається максимально розділова гіперплощина, а відповідний їй лінійний класифікатор називається максимально розділовим класифікатором[8].

Переваги:

- Дуже висока стійкість до перенавчання.
- Можливість зменшення чутливості до шуму за рахунок зниження точності. Однак пошук оптимального відношення даних параметрів потребує точного налаштування.

- Висока швидкість роботи в порівнянні з нейронними мережами.

Недоліки:

- Точність роботи метода поступається таким методам як AdaBoost і SNoW.

Одне і те ж обличчя, виділене різними алгоритмами, має різний розмір. Так алгоритм на базі бустинга виділяє обличчя повністю, захоплюючи лоб, підборіддя та щоки. Алгоритми на базі Snow та опорних векторів виділяють тільки очі, ніс і рот, причому алгоритм на базі опорних векторів виділяє обличчя вузьким вікном. Ці відмінності викликані тим, що в процесі створення алгоритмів використовувалися різні набори навчальних зображень для побудови класифікатора.

Порівняння точності знаходжень і розміру помилки другого роду при використанні описаних методів представлено в табл. 1[9].

Таблиця 1 – порівняння ефективності методів виявлення облич

Метод	Відсоток правильних знаходжень	Помилка другого роду
Нейронні мережі	~92%	~1.3%
Метод опорних векторів	~72%	~0.6%
SNoW	~94%	~0.12%
AdaBoost	~94%	~0.00001%

Як видно з представлених даних, найкращим за показниками відсотка вірних виявлень і помилки другого роду є алгоритм AdaBoost. Тому при створенні системи розпізнавання осіб в відеопотоках будемо використовувати метод Віоли-Джонса, заснований на даному алгоритмі.

Крім переваг в точності виявлення, даний метод має високу швидкість роботи, що робить його найбільш підходящим для виявлення осіб в реальному часі.

1.2. Методи розпізнавання облич

Методи розпізнавання осіб можна розділити на дві підгрупи. Перша підгрупа - це методи, засновані на значеннях пікселів, і методи, засновані на характерних точках[10]. Розглянемо основні методи кожної з цих підгруп.

1.2.1. Методи, засновані на значеннях яскравості пікселів

Назва даної групи методів означає, що для розпізнавання виявлених облич використовується тільки колір чи яскравість пікселів на зображеннях. Найпростішим схожим методом є порівняння, в якому мірою схожості є відстань між векторами яскравості пікселів зображень. Однак даний метод абсолютно не стійкий до змін освітленості, положення обличчя, масштабування. Більш того, такий підхід має високу обчислювальну

складність і абсолютно не придатний для розпізнавання в реальному часі. Тому, дуже часто використовуються методи, які використовують перехід векторного опису зображень в простору з меншою розмірністю, в яких порівняння набагато ефективніше. [10]

Eigenfaces - алгоритм, запропонований в 1991 році Метью Терком і Алексом Пентланд [11], який здобув широку популярність в якості першого успішного методу розпізнавання осіб. Основною ідеєю алгоритму є застосування методу головних компонент для знаходження векторів, найкращим чином описують зображення осіб. Використовуючи цей метод можна виявити різні зміни в навчальній вибірці зображень облич і описати цю зміну в базисі декількох ортогональних векторів, які називаються власними (eigenface). Обличчя, мають корисну властивість: зображення, яке відповідає кожному вектору має форму обличчя. Обчислення головних компонент зводиться до обчисленню власних векторів і власних значень коваріантної матриці, яка розраховується із зображення. Отриманий один раз на навчальній вибірці зображень облич набір власних векторів використовується для кодування всіх інших зображень осіб, які представляються зваженої комбінацією цих власних векторів. Використовуючи обмежену кількість власних векторів можна отримати стислу апроксимацію вхідному зображенню особи, яку потім можна зберігати в базі даних у вигляді вектору коефіцієнтів, який служить одночасно ключем пошуку в базі даних осіб [11]. Сума головних компонент помножених на відповідні власні вектори являється реконструкцією зображення.

Для кожного зображення обличчя обчислюється його головні компоненти. Зазвичай це від 5 до 200 головних компонент. Процес розпізнавання ґрунтується на порівнянні головних компонент невідомого зображення з компонентами всіх відомих зображень.

При цьому робиться припущення, що зображення облич, які відповідають одній людині, згруповані в кластери у власному просторі. З

бази даних вибираються зображення-кандидати, які мають найменшу відстань від вхідного зображення.

Метод власних обличч потребує для свого застосування ідеалізування умов таких як, єдині параметри освітленості, нейтральний вираз обличчя, відсутність завад в якості окулярів і бороди. При недотриманні цих умов головні компоненти не будуть відображати міжкласові варіації. Наприклад, при різних умовах освітленості метод власних обличч майже не застосовується, оскільки перші головні компоненти переважно відображають зміни освітленості, і порівняння видає зображення, яке має схожий рівень освітленості. При дотриманні ідеалізованих умов точності розпізнавання з використанням методу досягає більше 90%.

Обчислення набору власних векторів відрізняється високою трудомісткістю. Один із способів – верстка зображень по стрічках та стовпчикам; в такій формі представлення зображення має на порядок менший розмір, обчислення і розпізнавання проходить швидше, але відновити початкове зображення неможливо.

Метод головних компонент добре зарекомендував себе в практичних додатках. Однак, в тих випадках, коли на зображенні особи присутні значні зміни в освітленості або виразі обличчя, ефективність методу значно падає.

Ідея методу складається з представлення зображення обличчя у вигляді набору векторів головних компонент зображень, так званих

Переваги:

- Простоту реалізації,
- Придатність для розпізнавання в реальному часі
- Можливість компактно зберігати великі об'єми даних. [10]
- Швидке розпізнавання

Недоліки:

- Обчислювання набору власних векторів вирізняється високою трудомісткістю.

- Відсутність таких перешкод, як окуляри або борода.
- Повинна бути проведена якісна попередня обробка, що приводить зображення до стандартних умов.
- Зображення повинні бути отримані в близьких умовах освітленості, однаковому ракурсі.
- Навчання проходить один раз, при додаванні нових облич в базу, прийдеться будувати простів з власних векторів заново.

Недоліки викликані тим, що найбільш важливі власні вектори в більшій мірі описують особливості освітлення, ніж характеристики облич, оскільки спочатку метод головних компонент вибирає підпростір з метою апроксимації даних, а не їх класифікації [10].

Fisherfaces - алгоритм, в якому на відміну від методу eigenfaces використовується лінійний дискримінантний аналіз, а саме лінійний дискримінант Фішера. Дія алгоритму заснована на пошуку проєкції даних, при якій класи зображень облич максимально роздільні. При використанні методу головних компонент проводиться максимізація розкиду даних по всій базі облич. Ця відмінність дозволяє вирішити проблему високої чутливості до змін освітлення [12].

Метод головних компонент, який являється основою методу Eigenfaces, знаходить лінійну комбінацію ознак, яка максимізує загальну дисперсію даних. Це доволі сильний спосіб представлення даних, але він не враховує класи об'єктів, і може бути втрачено багато відмінної інформації через відкидання компонент, які не пройшли базис. Наприклад, коли в базі є фотографія з різним освітленням, і оскільки відомо, що компоненти, визначені за допомогою методу головних компонент, не завжди містять в собі всю відмінну інформацію, тому зразки різних класів змішались один з одним, і подальша класифікація стає неможливою. Цю проблему рішає алгоритм лінійного дискримінантного аналізу, який є основою методу Fisherfaces.

Лінійний дискримінантний аналіз – метод статистики машинного навчання, який застосовується для пошуку лінійних комбінацій ознак, які найкращим чином розділяють два чи більше класи об'єктів чи подій (тобто об'єкти одного і того ж класу мають шукатись, як можна ближче один до одного в просторі, і при цьому відбувається максимізація відстані між класами).

Даний алгоритм краще розпізнає при різному освітленні, чим EigenFaces, тільки навчання має відбуватись також при різних умовах освітленості.

Переваги:

- За рахунок множини фотографій кожної персони алгоритм виходить стійким до змін умов освітленості.

Недоліки:

- Відсутність інваріантності щодо афінних перетворень.

Локальні бінарні шаблони (надалі LBP - Local Binary Pattern) - простий і ефективний оператор перетворення зображень, вперше запропонований в 1996 році для класифікації текстур. [13] Однак, пізніше знайшов застосування і для розпізнавання осіб. [14]

Даний оператор використовує значення яскравостей околиці кожного пікселя зображення і за допомогою функції присвоює кожному пікселю значення, яке описує його околиці. Далі отримане зображення розділяється на підобласті, для кожної з яких розраховується гістограма. Гістограми конкатенуються і порівнюються за допомогою методів машинного навчання. У класичному варіанті використовується метод найближчого сусіда. [14]

Переваги цього методу полягають в простоті реалізації і високої швидкості роботи, яку можна збільшити, використовуючи різні модифікації алгоритму. При цьому алгоритм показує високі результати при розпізнаванні облич і стійкий до монотонних змін освітлення. Все це робить

його оптимальний для розпізнавання облич в системах обробки в реальному часі.

1.2.2. Методи, засновані на характерних точках

Дана група методів, на відміну від попередньої, використовує характерні точки і їх координати на зображенні, а не оцінює яскравість пікселів. Такими характерними точками можуть бути, наприклад, центри очей, положення носа, лінія брів, рота і т. д. [10]. До даного класу методів відносяться активні моделі зовнішнього вигляду і активні моделі форми.

Активні моделі зовнішнього вигляду (Active Appearance Models, ААМ) - це статистичні моделі зображень, які шляхом різного роду деформацій можуть бути підігнані під реальне зображення. Даний тип моделей в двовимірному варіанті було запропоновано Тімом Кутсом і Крісом Тейлором в 1998 році. [15] Активна модель зовнішнього вигляду містить два типи параметрів: параметри, пов'язані з формою (параметри форми), і параметри, пов'язані зі статистичною моделлю пікселів зображення або текстурою (параметри зовнішнього вигляду). Перед використанням модель повинна бути навчена на безлічі заздалегідь розмічених зображень. Розмітка зображень виробляється вручну. Завдяки використанню активної моделі наявність перешкод, як, наприклад, вуса, система в частині випадків зможе дати відповідь, оскільки відхилення значень на окремих точках може не впливати на загальну відповідь системи. Спочатку здійснюється локалізація обличчя за допомогою маски обличчя. Потім виконується трекінг обличчя, використовуючи знайдену позицію та розмір за допомогою активного контуру моделі та відповідно із цієї моделі отримуються ключові точки.

Активні моделі форми (Active Shape Models, ASM) враховують статистичні зв'язки у взаємному розташуванні антропометричних точок. На кожному зображенні вибірки експерт розмічає розташування

антропометричних точок. Для того щоб привести координати на всіх зображеннях до єдиної системи зазвичай виконується т. зв. узагальнений Прокрустом аналіз, в результаті якого всі точки приводяться до одного масштабу і центруються. Далі для всього набору образів обчислюється середня форма і матриця коваріації. На основі матриці коваріації обчислюються власні вектори, які потім сортуються в порядку спадання відповідних їм власних значень. Локалізації ASM моделі на новому, що не входить в навчальну вибірку зображенні здійснюється в процесі рішення оптимізаційної задачі [16].

Однак варто зазначити, що подібні моделі спочатку призначені не для розпізнавання, а для точної локалізації характерних точок на зображеннях облич. Їх локалізація дозволить виконати процедуру вирівнювання облич вибірки і приведення їх до однієї системи координат для більш точного розпізнавання іншими методами. Зазвичай для цих цілей використовується невелика кількість точок, що дозволяє прискорити продуктивність. Для задач розпізнавання, навпаки, потрібна велика кількість характерних точок, що збільшить точність класифікації і знизити швидкість роботи системи [17].

В результаті аналізу можна відзначити, що оптимальним методом для обробки в реальному часі є метод з використанням локальних бінарних шаблонів. У наступних розділах даний метод буде описаний більш детально, а також буде наведено дослідження продуктивності різних модифікацій методу, яке дозволить виявити найкращу для вирішення поставленого завдання модифікацію з точки зору співвідношення швидкості роботи і точності класифікації.

1.3 Аналіз задачі розпізнавання облич у відеопотоках

Як було зазначено раніше, обробка кадрів відеопотоку розроблюваною системою повинна включати два основних етапи. Перший етап - виявлення осіб методом Віоли-Джонса.

Другий етап - розпізнавання знайдених осіб за допомогою гістограм локальних бінарних шаблонів і методу найближчого сусіда. Однак продуктивність даних алгоритмів істотно залежить від таких факторів як освітлення, положення особи і т. д.. Тому, доцільно відразу описати умови застосування системи, що розробляється, в яких може бути забезпечена її коректна робота:

- Допустимо тільки монотонне зміна освітлення. Навчальна та тестова вибірка повинна зніматися в однакових умовах освітлення.
- Використовується фронтальне, або близькі до нього положення осіб. Нейтральний вираз облич в зображеннях.
- Особи не перекриваються іншими об'єктами.

Також доцільно відразу описати необхідний функціонал, що розробляється:

- Обробка відеопотоку з підключеною до комп'ютера камери в реальному часі.
- Можливість налаштування параметрів роботи використовуваних для виявлення і розпізнавання алгоритмів.
- Виведення інформації про обличчя, що розпізнається, що включає міру належності до певного класу, графічне відображення гістограми і LBP представлення обличчя, що відслідковується.
- Можливість навчання і додавання класів осіб з використанням камери через інтерфейс програми.

Крім етапів виявлення і розпізнавання доцільно використовувати проміжні етапи обробки знайдених осіб. Застосування фільтра Гауса після

виявлення осіб допоможе знизити вплив шумів при розпізнаванні. Також є сенс застосувати маску значущих областей до локалізованих і перетворених оператором LBP зображенням осіб, яка дозволить прибрати вплив при розпізнаванні кутових областей зображення, що містять задній план.

В результаті загальний алгоритм розпізнавання повинен містити наступні кроки: виявлення осіб в кадрі, обробка знайдених осіб фільтром Гауса, застосування LBP трансформації до знайденим особам з подальшим застосуванням маски значущих областей, розрахунок гістограм знайдених осіб, класифікація осіб по гістограмі методом найближчого сусіда. В результаті буде отримано список що відслідкованих облич з їх характеристиками і координатами прямокутних областей кадру, в яких вони знаходяться.

Узагальнена блок-схема алгоритму обробки кадрів розробляється системою представлена на (рис. 1).



Рисунок 1 - Узагальнена блок-схема алгоритму обробки кадрів відеопотоку

Процедура навчання проводиться аналогічним чином. Знайдене в кадрі обличчя послідовно обробляється відповідно до описаного алгоритмом, розраховані гістограми осіб навчальної вибірки кожного класу зберігаються. Розпізнавання проводиться на основі пошуку мінімальної відстані між гістограмою вхідного зображення особи і гістограм, що зберігаються базі.

Висновки

1. На основі аналізу методів виявлення облич, порівняння їх переваг і недоліків, обґрунтовано вибір методу Віюлі-Джонса в якості алгоритму виявлення облич в розроблюваній системі.

2. На основі аналізу методів розпізнавання облич, порівняння їх переваг і недоліків, обґрунтовано вибір методу локальний бінарних шаблонів в якості алгоритму розпізнавання облич, що має бути покладений в основу розроблюваної системи.

3. Проаналізовано завдання та виявлено основні проблеми, які можуть виникнути в ході її вирішення.

4. Прийнято умови застосування системи, що розробляється, необхідні для її коректної роботи.

5. Описано функціональні вимоги до розроблюваної системи.

6. Описано узагальнений алгоритм обробки кадрів відеопотоку

2. ЕТАПИ ОБРОБКИ КАДРІВ ВІДЕОПОТОКУ

У даному розділі будуть детально описані етапи обробки кадрів відеопотку, а також методи та алгоритми, які будуть використовуватись.

2.1. Виявлення облич методом Віоли-Джонса

Як відмічалось раніше, даний метод був розроблений та представлений Полом Віолою і Майклом Джонсом в 2001 році[3]. Однак і по цей день, даним метод являється одним із основних методів пошуку об'єктів на зображенні в реальному часі.

Виділимо основні принципи, на яких основана робота даного методу:

- Інтегральне представлення зображень.
- Пошук облич за допомогою ознак Хаара.
- Каскадна класифікація з застосування бустингу.

2.1.1 Виявлення облич методом Віоли-Джонса

Для розрахунку яскравості прямокутної області зображення використовується інтегральне представлення[18]. Дане представлення часто застосовується і в багатьох розроблених алгоритмах комп'ютерного зору. Інтегральне представлення дозволяє швидко розрахувати загальну яскравість довільного прямокутника на заданому зображенні, при чому час розрахунку не залежить від площі прямокутника.

Інтегральне представлення зображення являє собою матрицю, розміри якої співпадають з розмірами вхідного зображення. У кожному елементі такої матриці зберігається сума інтенсивностей таких пікселів, які знаходяться лівіше і вище даного елемента. Елементи матриці розраховуються у відповідності з наступною формулою:

$$I(x, y) = \sum_{x' \leq x, y' \leq y} i(x', y')$$

Де $I(x,y)$ – значення точки (x,y) інтегрального зображення; $i(x,y)$ – значення інтенсивності вхідного зображення.

Застосування інтегрального представлення зображення дозволяє обчислювати ознаки однакового виду, але з різними геометричними параметрами, за однаковий час, так як розрахунок матриці інтегрального представлення займає лінійний час, пропорційне числу пікселів у зображенні.

2.1.2 Ознаки Хаара

Ознакою f об'єкта a називають відображення $f:A \rightarrow D_f$, де D_f - множина допустимих значень ознаки. Якщо заданий набір векторів f_1, \dots, f_n , тоді вектор $x=(f_1(a), \dots, f_n(a))$ називається ознаковим описом об'єкта $a \in A$ [19].

Вперше використання для виявлення об'єктів ознак заснованих на вейвлетах Хаара, було запропоноване в роботі Папагоргиу в 1998 році [20]. Віола і Джонс адаптували цю ідею в своїй роботі і отримали прямокутні ознаки, названі ознаками Хаара[3]. Зовнішній вид ознак можна побачити на рис. 2.

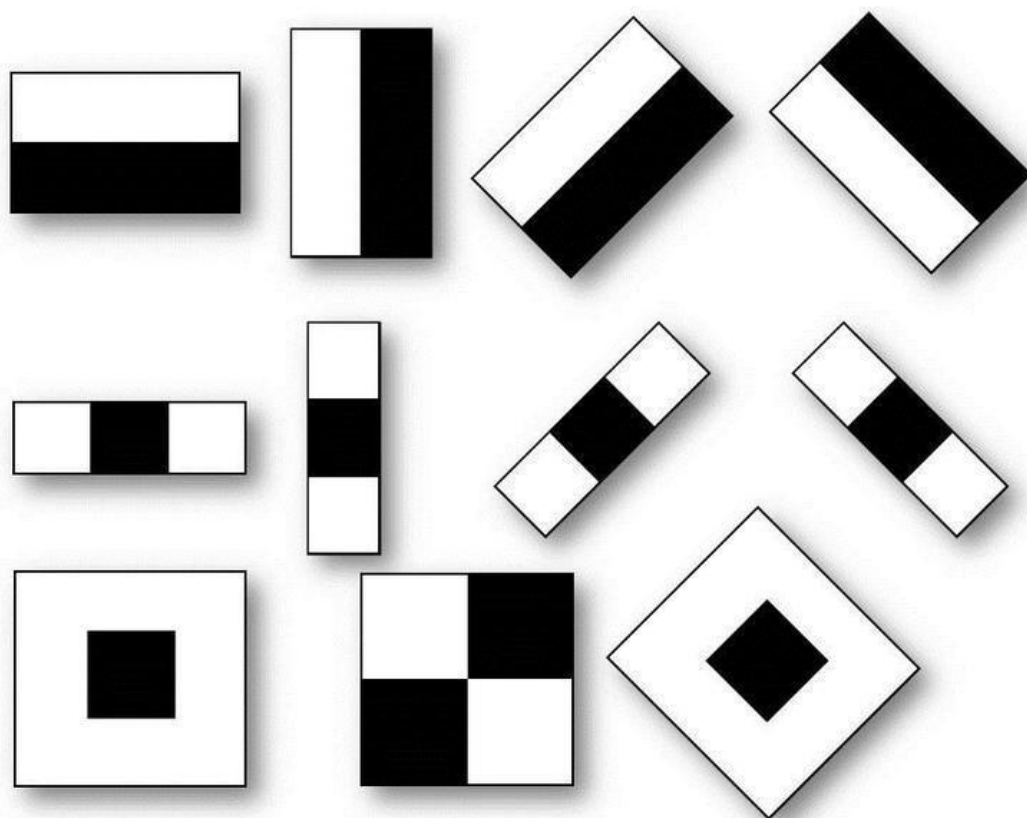


Рисунок 2 – Ознаки Хаара

У розширеному методі Віоли-Джонса, представленому в бібліотеці комп'ютерного зору OpenCV, і використовуваному у розроблюваній системі, використовується також додаткові ознаки, представлені на (рис. 3).

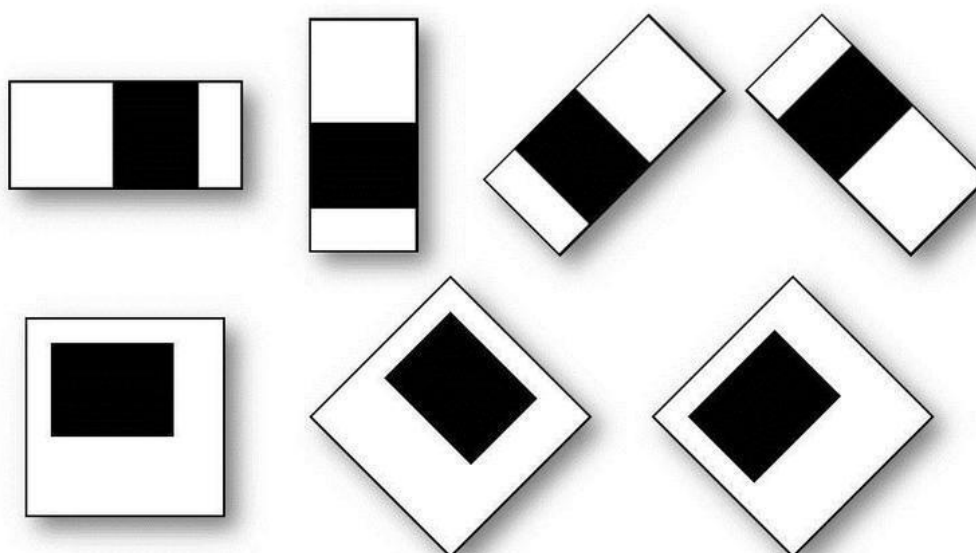


Рисунок 3 – Додаткові ознаки Хаара

Результатом обчислення такої ознаки на інтегральному представленні буде:

$$F = U - V$$

Де U – сума значень яскравостей точок, закритий світлою частиною ознаки, а V – сума значень яскравостей точок, закритих темною частиною ознаки. Такі ознаки описують перепад яскравості по двом осям зображення.

Пошук облич виконується за допомогою так званого скануючого вікна, розміри якого в оригінальному алгоритмі складають 24×24 пікселя. Вікно рухається по зображенню з кроком в 1 піксель і для кожного його положення обчислюються ознаки Хаара з різним масштабом і положенням у вікні. При цьому саме сканування проводиться так само і для різних масштабів скануючого вікна. Знайдені ознаки передаються класифікатору, який визначають по їх значенням, чи являється область зображення, яка відноситься до вікну, обличчям чи ні.

2.1.3 Каскадна класифікація

Каскадна структура класифікатора дозволяє пришвидшити виявлення облич, фокусуючи роботу на найбільш цікавих областях зображення. Каскад надає собою структурну організацію слабких класифікаторів, навчених з застосуванням процедури бустингу. Таким чином при малих обчислювальних витратах можна на ранніх етапах розпізнавання відкинути зображення, з великою долею імовірності, які не містять шуканий об'єкт (в даному випадку обличчя). Приклад каскадної структури класифікаторів представлений на (рис. 4).

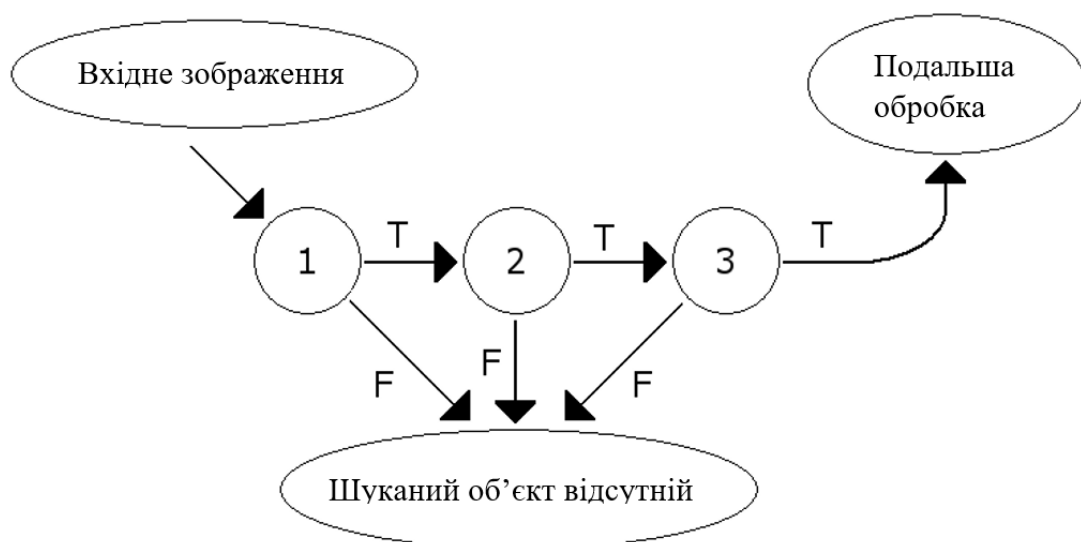


Рисунок 4 – Каскадний класифікатор

Кожний рівень каскаду навчається за допомогою раніше згаданого алгоритму AdaBoost[3]. Кількість особливостей, використовуваних в ньому, збільшуватись до тих пір, поки виявлення цільового об'єкта і помилки першого роду не досягнуть заданого значення. Рівні визначається шляхом тестування поточний детектора на множині, що перевіряється. Якщо загальна помилка першого роду для всього об'єкта ще не досягнута, то в каскад додається ще один шар. Від'ємна множина для навчання наступних шарів визначається шляхом збору всіх хибних виявлень при використанні поточного каскаду.

У результаті класифікації буде отриманий набір областей зображення, якій містять шуканий об'єкт. Потім виключаються вкладені повторення при виявленні одного і того ж об'єкта, які викликані масштабуванням скануючого вікна.

Для подальшої обробки знайдені обличчя переводяться в градації сірого і масштабуються до розміру 128*128 пікселів.

2.2. Фільтр Гауса

З ціллю усунення шумів на зображеннях обличч використовується фільтр Гауса. Фільтр Гауса – це фільтр розмиття зображення, який

використовує нормальне розподілення для обчислення перетворення, яке застосовується до кожного пікселя зображення. Нормальне розподілення для двох вимірів описується формулою[21]:

$$G(u, v) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/(2\sigma^2)}$$

Де r – радіус розмиття; $r = x^2+y^2$; σ – стандартне відхилення розподілення Гауса.

Дана формула задає поверхню, яка має вигляд концентрованих окружностей з нормальним розподіленням від центральної точки. Пікселі, де розподілення відмінне від нуля використовується для побудови матриці згортки, яка застосовується до вхідного зображення. Значення пікселя приймає найбільшу вагу (має найвище Гаусове значення), і сусідні пікселі приймають менші ваги, в залежності від відстані до них.

Приклад роботи розмиття по Гаусу для одномірного масиву можна побачити на (рис. 5).

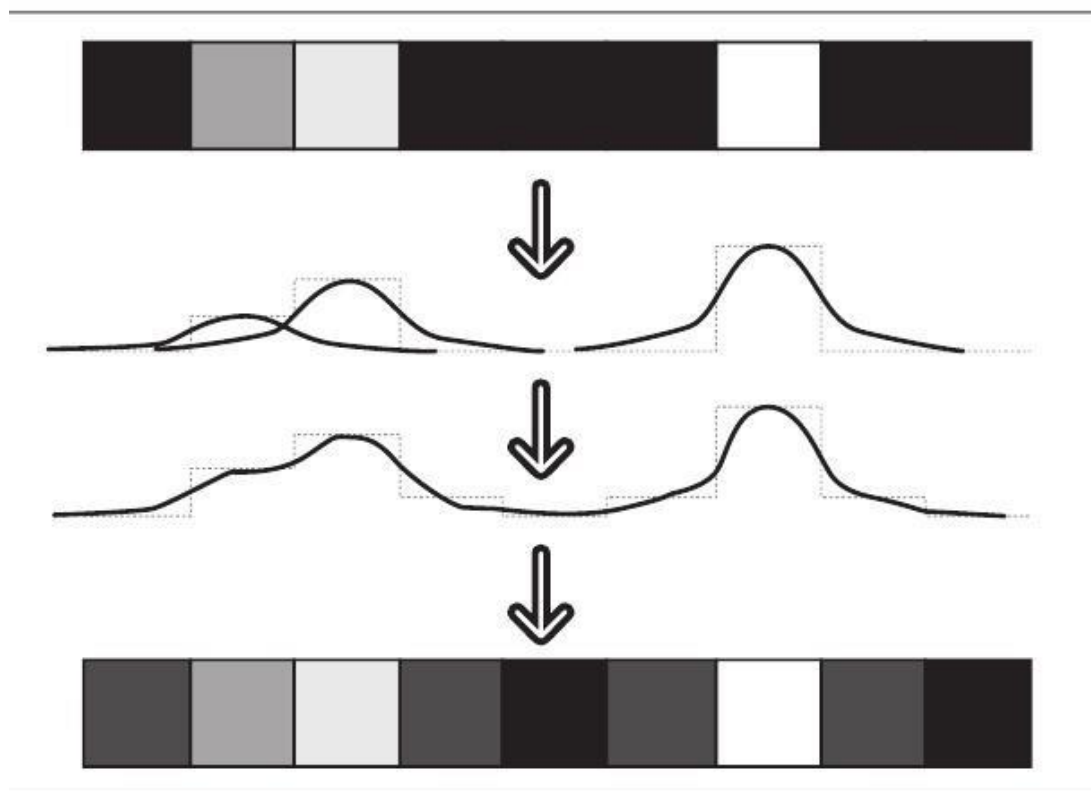


Рисунок 5 – Розмиття по Гаусу на одномірному масиві

Таким чином, розмиття по Гаусу дозволяє нам позбутися від небажаних шумів на зображеннях, що зведе до мінімуму їх вплив при подальшій класифікації облич. Результат застосування фільтра Гауса до цілого зображення продемонстрована на (рис. 6).



Рисунок 6 – Результат застосування фільтра Гауса.

2.3. LBP перетворення

2.3.1. Класичний алгоритм

LBP оператор вперше був запропонований в 1996 році для класифікації текстур[13]. Однак, пізніше знайшов застосування і для розпізнавання облич[14]. Суть оператора ґрунтується в застосування до пікселів зображення граничного перетворення, в якому значення яскравості у пікселя, що обробляється, порівнюється зі значенням яскравості пікселів

його околиці. Результат порівняння кожного пікселя околиці з пікселем, що оброблюється конкатенується у двійкове число.

У класичному варіанті використовується квадратна околиця 3*3 пікселя. Приклад розрахунку LBP перетворення для такої околиці наведено на (рис. 7).

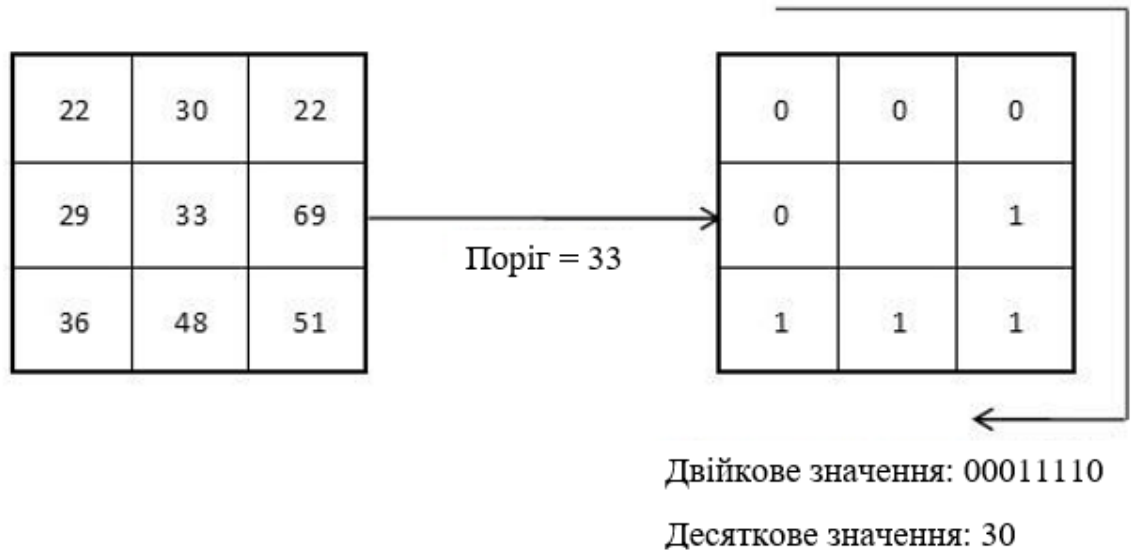


Рисунок 7 – Класичний LBP оператор.

Після застосування LBP оператора, зображення ділиться на прямокутні області, для кожної з яких розраховується гістограми, які описуються наскільки часто зустрічаються в даній області пікселі різної яскравості. Значення елементів LBP гістограми можуть бути описані наступною формулою:

$$H_i = \sum_{x,y} I\{f(x,y) = i\}, i = 0, \dots, n - 1$$

Де $f(x,y)$ – значення яскравості пікселя LBP зображення з координатам (x,y) ; n – кількість різних значень яскравостей пікселів; $I\{A\}=1$, якщо A – істина, інакше $I\{A\}=0$.

Отримані гістограми нормалізуються, конкатенуються і використовуються в подальшому в якості ознак класифікації. В оригінальному дослідженні для класифікації використовується метод

найближчого сусіда, який детально буде описаний в наступних розділах. Приклад розбиття зображення на прямокутні області і формування гістограм можна побачити на (рис. 8).

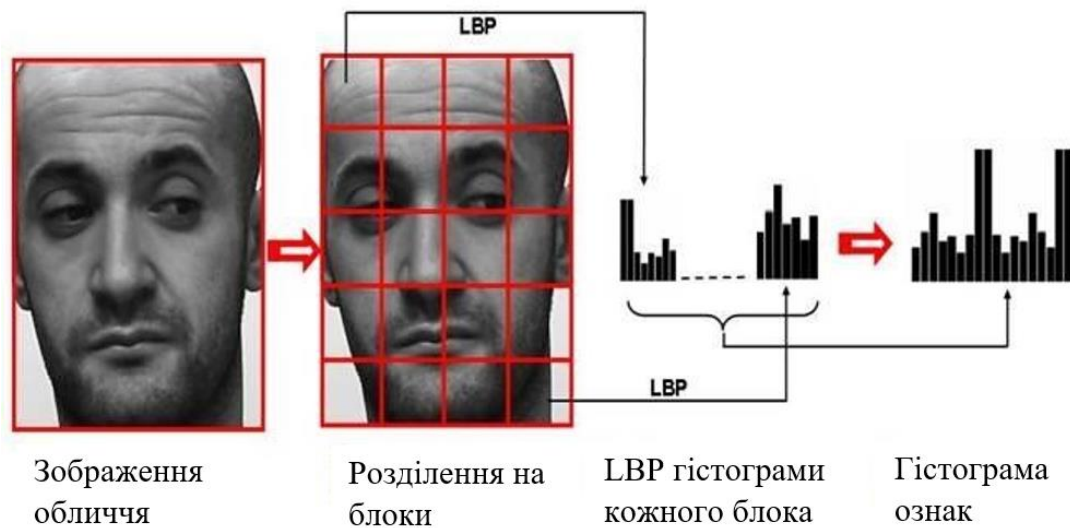


Рисунок 8 – Розбиття зображення на прямокутні області формування гістограм

В результаті виходить опис зображення обличчя на трьох рівнях локалізації. При цьому такий опис не залежить від монотонних змін освітлення.

2.3.2. Рівномірні LBP

Подальші дослідження локальних бінарних шаблонів показали, що суттєву інформацію про форму об'єктів на зображенні несе тільки частина з них[21]. Такі локальні бінарні шаблони були названі рівномірними.

До даного виду LBP відносяться ті шаблони, двійковий код яких містить не більше двох переходів між нулем та одиницею. Вони описуються лише важливі локальні особливості зображення, такі як кінці ліній, грані, плями. Приклади рівномірних LBP представлені на (рис. 9).

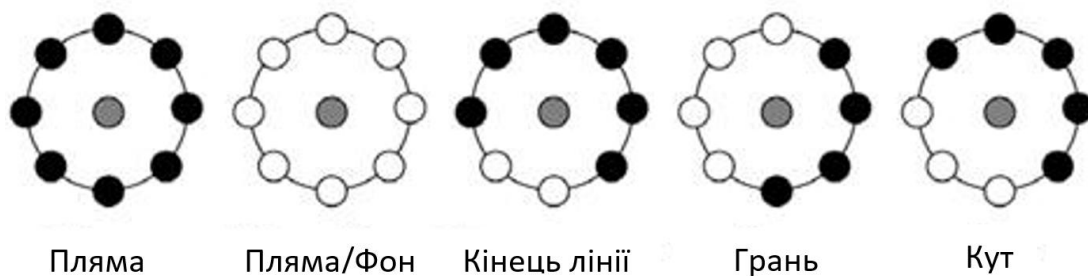


Рисунок 9 – Рівномірні LBP.

Всього налічується 58 рівномірних LBP. В результаті виходить 59-мірна гістограма ознак (додатковий розряд відводиться для підрахунку всіх нерівномірних LBP), у відмінності від 256-мірної гістограми в оригінальному алгоритмі. Таке скорочення розмірності дозволяє знизити витрати пам'яті і суттєво збільшити швидкість класифікації, при цьому покращивши її показники за рахунок використання тільки важливих ознак.

2.3.3. Центральнo-симетричні LBP

Дана модифікація алгоритму розрахунку локальних бінарних шаблонів дозволяє ще сильні скоротити витрати пам'яті і обчислювальну складність класифікації. Суть модифікації оснований на тому, що в якості граничного значення для кожного пікселя околиці приймається не значення яскравості центрального пікселя околиця, а значення яскравості протилежного відносно центра околиці пікселя[23]. Порівняння розрахунку класичного LBP і центральнo-симетричного LBP наведено на (рис. 10).

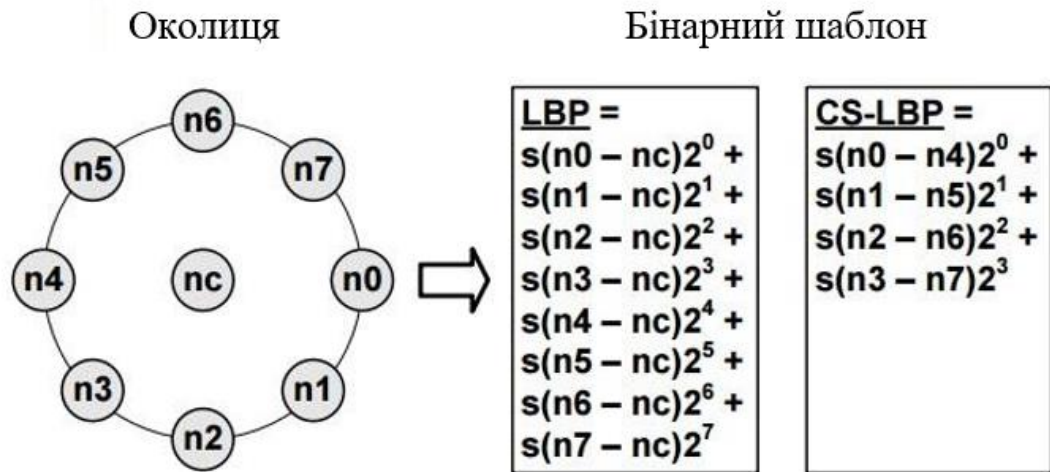


Рисунок 10 – Розрахунок значення звичайного і центрально-симетричного LBP (CS-LBP).

Як бачимо, в даному алгоритмі число розрядів значень перетворень пікселів скорочується до чотирьох. Відповідно розмірність гістограми ознак зменшується до $2^4 = 16$. В результаті отримуємо ще більшу економію пам'яті і збільшення швидкості класифікації чим при використанні рівномірних LBP. Ці переваги роблять дану модифікацію алгоритму ідеальним вибором для класифікації в реальному часі.

Однак подібне скорочення розмірності гістограми ознак може негативно вплинути на точність класифікації. Відповідно при виборі ознак класифікації необхідно знайти компроміс між швидкістю роботи і точністю. Дослідженню продуктивності вищеповисаних алгоритмів присвячений окремий розділ цієї роботи.

2.4. Маска значимих областей зображення

Зображення облич, які отримуються після процедури виявлення, мають квадратну форму. Однак обличчя займає не весь простір зображення. Тому логічно було б виключити вплив на рішення класифікатора областей зображення, в яких немає обличчя.

Простим способом рішення такої проблеми є застосування маски значимих областей зображення. Така маска представляє собою зображення однакового розміру з вхідним зображенням. Пікселі ненульової яскравості в масці відповідає значимим областям. В нашому випадку значимою областю є овальна область в центрі зображення, яка відповідає обличчю.

Значення пікселів результуючого зображення можна визначити за формулою:

$$R(x, y) = \begin{cases} I(x, y) & \text{якщо } M(x, y) \neq 0 \\ 0 & \text{якщо } M(x, y) = 0 \end{cases}$$

де $I(x, y)$ – значення яскравості пікселя оброблюваного зображення;
 $M(x, y)$ – значення яскравості пікселя маски значимих областей.

При рішенні задачі класифікації з використанням локальних бінарних шаблонів маску значимих областей доцільно застосувати після виконання LBP перетворення і перед розрахунком гістограм. Таким чином, всі незначимі пікселі зображення на гістограмі будуть згруповані в одне значення. Приклад застосування маски значимих областей до зображення, обробленого LBP оператором, наведено на (рис 11).

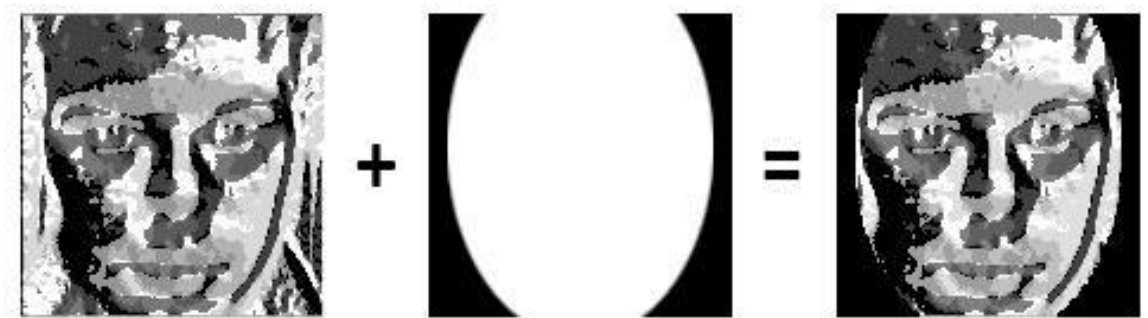


Рисунок 11 – Застосування маски значимих областей.

З рисунку видно, що в результаті застосування даної операції пікселі зображення, які не мають впливати на результат класифікації, приймають нульове значення яскравості.

2.5. Метод найближчого сусіда

Отримані LBP гістограми класифікуються методом найближчого сусіда, як і в оригінальному дослідженні LBP стосовно до задачі розпізнавання облич[14]. Метод найближчого сусіда є простим алгоритмом класифікації, суть якого полягає в тому, що об'єкт належить до того класу, до елемента якого він найближче знаходиться. Наприклад, на (рис. 12) коло зеленого кольору відповідно до цього алгоритму повинне бути класифіковане як червоний трикутник.

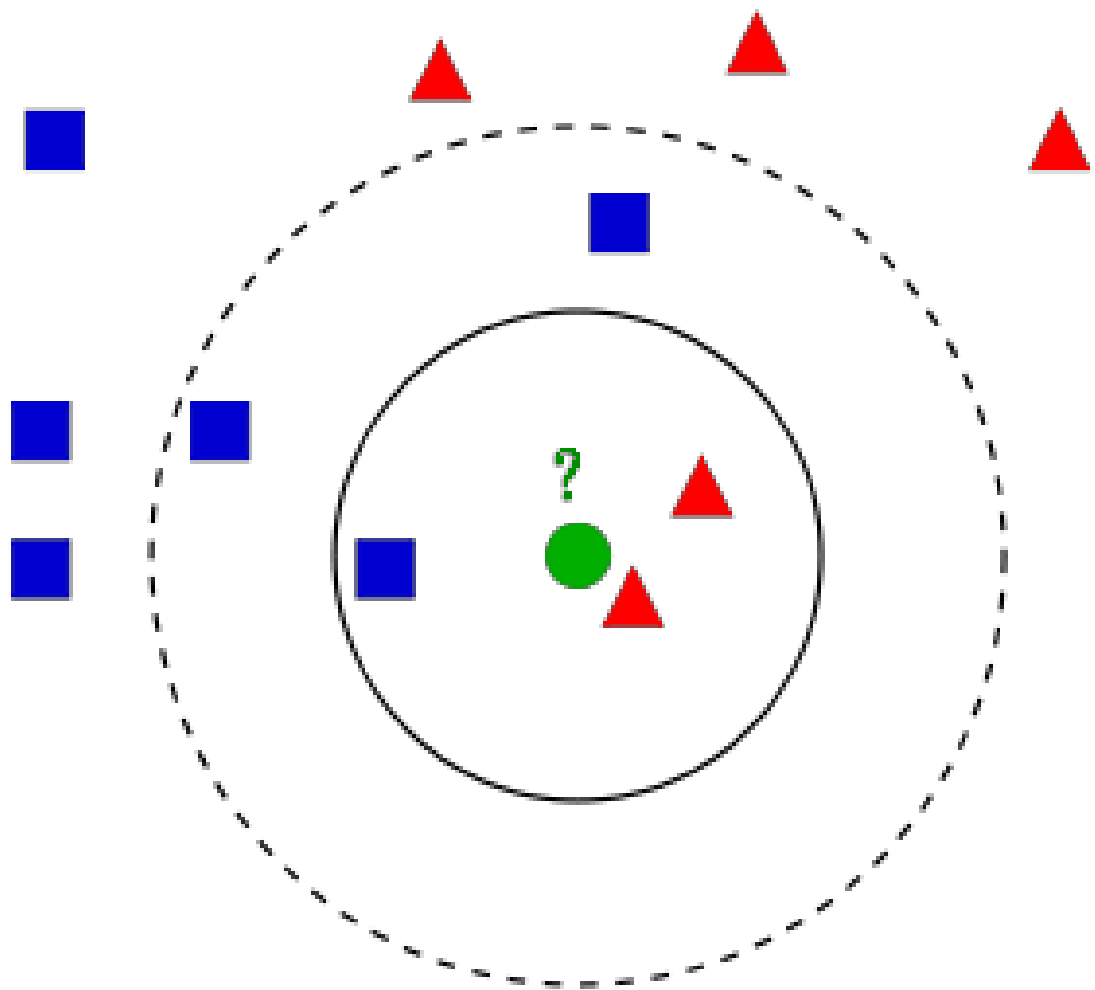


Рисунок 12 – Метод найближчого сусіда

Метод, в якому об'єкт відносять до класу, до якого відносяться більшість його сусідів в околиці заданого розміру, використовується для покращення результатів. Однак дослідження, проведене в рамках даної

роботи, показало, що при рішенні задачі класифікації осіб такий підхід негативно впливає на роботу класифікатора.

Математично алгоритм можна описати таким чином. На першому кроці визначається елемент x_s навчальної вибірки з N елементів, який найближче до представленого образу x , тобто:

$$\|x - x_s\| = \min\{\|x - x_i\| : i=1, \dots, N\}$$

На другому кроці перевіряється умова належності до класу: якщо $x_s \in \omega_i$, то тоді рахується, що $x \in \omega_i$ [26].

Даний метод застосовують в тому випадку, коли ціна помилки неправильної класифікації є велика, а помилки даних невеликі. Основним недоліком методу найближчого сусіда є його чутливість до значень окремих (можливо помилкових) даних. Незважаючи на це, метод показує високу ефективність при застосуванні в широкому спектрі завдань класифікації [24].

Особливу увагу також заслуговує питання вибору метрики, яка визначає відстань між гістограмами. Для досягнення максимальної точності класифікації необхідно вибрати ту метрику, яка найбільш адекватно б відображала відмінності між гістограмами зображень різних класів. В оригінальному дослідженні [14] застосовується так звана відстань Chi-Square, яка розраховується по наступній формулі.

$$\sum_{i=1}^n \frac{(x_i - y_i)^2}{(x_i + y_i)}$$

де x_i – i -те значення першої гістограми; y_i – i -е значення другої гістограми.

Висновки

1. Описаний алгоритм виявлення облич за допомогою метода Віюлі-Джонса.
2. Описаний принцип роботи фільтра Гауса, який застосовується для зменшення впливу шумів на результат класифікації.

3. Описаний механізм роботи LBP оператора і його різних модифікацій, а також принцип розрахунку LBP гістограм.
4. Описаний етап застосування маски значимих областей, які дозволяють виключити вплив на розпізнавання областей зображення, які не містять обличчя.
5. Описана класифікація гістограм методом найближчого сусіда і принцип розрахунку відстаней між гістограмами.

3. ДОСЛІДЖЕННЯ ЕФЕКТИВНОСТІ LBP ОПЕРАТОРІВ

Як вже було зазначено раніше, з точки зору швидкодії ідеальним вибором в якості ознак класифікації при розробці системи розпізнавання в реальному часі є центрально-симетричні локальні бінарні шаблони. Проте, перед застосуванням даного алгоритму в розроблюваній системі є сенс переконатися в тому, що він не сильно програє більш повільним варіаціям LBP в точності класифікації.

3.1. Методика тестування

Для вирішення даного завдання всі три описаних раніше алгоритму розрахунку LBP гістограм були протестовані на двох різних наборах даних. При цьому також оцінювався найкращий варіант розбиття класифікованих зображень на локальні області. В якості алгоритму класифікації використовувався метод найближчого сусіда. Створені в процесі дослідження реалізації LBP перетворень і методу найближчого сусіда в подальшому були застосовані при розробці системи розпізнавання осіб в відеопотоках, а також при тестуванні швидкості роботи цієї системи з використанням різних варіацій LBP перетворення.

Перший набір даних, використаний для тестування - це база зображень облич лабораторії Кембриджського Університету[25]. Вона містить зображення 40 осіб, по 10 зображень на кожного. Освітлення на даних зображеннях не змінюється, однак присутні варіації в положенні особи при зйомці. Зображення одного з осіб даної бази представлені на (рис. 13).



Рисунок 13 – Зображення з бази даних Кембриджського університету

В якості навчальної та тестової вибірки використовувалося по 5 зображень на кожну особу. Тестова та навчальна вибірка не мали перетинів, проте для отримання результатів на більш великому масиві даних тестова і навчальна вибірки мінялися місцями, і тестування проводилося повторно. Перед обробкою зображення масштабувалось до розміру 128×128 пікселів. В результаті всього було класифіковано 400 зображень.

Другий набір даних – база зображень облич лабораторії Йельського університету[26]. Дана база містить зображення 38 людей, по 65 зображень на кожного, включаючи різні зображення з освітленням. З них на тестування було відібрано по 10 зображень на кожну людину. Приклад зображень з другої бази облич наведений на (рис. 14).



Рисунок 14 – Зображення на базі даних Йельського університету

Варто відмітити, що в даній базі обличчя кадровані точніше і в більшому масштабі, чим на зображеннях з першого набору даних. Тестування проводилось аналогічно тестування на першому наборі. Всього було класифіковано 380 зображень.

3.2. Результати досліджень

Ефективність розпізнавання для кожного з трьох LBP операторів при тестування на першому наборі тестових даних представлена в (табл. 2).

Таблиця 2 – тестування LBP операторів на першому наборі даних

Розбиття на блоки Метод	1x1	2x2	3x3	4x4	5x5	6x6	7x7	8x8
LBP	82,5%	91%	94%	95,5%	94%	93,25%	92,5%	89,3%
Uniform LBP	81%	93,8%	97%	94,5%	92%	92%	92%	89,5%
CS-LBP	67,8%	92,3%	95%	94,8%	94,3%	93,3%	90,3%	90,3%

Ефективність розпізнавання для кожного з трьох LBP операторів при тестуванні на другому наборі тестових даних представлена в (табл. 3).

Таблиця 3 – тестування LBP операторів на другому наборі даних

Розбиття на блоки Метод	1x1	2x2	3x3	4x4	5x5	6x6	7x7	8x8
LBP	41,8%	71,6%	88,2%	91,8%	92,6%	93,2%	95,8%	96,1%
Uniform LBP	41,3%	75,26%	91,8%	91,8%	92,9%	92,1%	95%	94,2%
CS-LBP	20,3%	61,3%	84,5%	89,2%	89,2%	91,6%	92,6%	93,9%

Як видно з отриманих результатів, класичний LBP і Uniform LBP працюють приблизно з одною точністю. Для досягнення точності 90% і більше доцільно використовувати розбиття зображення починаючи від 4x4. Однак варто відмітити, що на першому наборі даних краще всього себе показало розбиття 3x3.

Центрально-симетричний LBP оператор при розбитті зображення на малу кількість блоків уступає іншим локальним бінарним шаблонам. Але при використанні великого числа підобластей його показник точності класифікації відстає від інших LBP в середньому не більше ніж на 3%. При тестуванні на першому наборі даних CS-LBP перевершує інші шаблони на ряді розбиття.

В результаті можна сказати, що центрально-симетричні локальні бінарні шаблони доцільно використовувати в системі розпізнавання обличч у відеопотоках, через високу швидкість роботи і показників точності, які майже не поступаються іншим LBP. Порівняння швидкості роботи системи розпізнавання при використанні всіх трьох варіацій LBP буде приведено в розділі, який присвячений тестуванню.

Оптимальним по співвідношенню точності і витрат пам'яті розбиття зображення на частини за допомогою CS-LBP являється розбиття 4x4, яке і буде використовуватись в розроблюваній системі.

Висновки.

1. Проведено дослідження ефективності розпізнавання облич при використанні трьох варіантів LBP перетворення: класичного, рівномірного і центрально-симетричного.
2. Результати досліджень показали, що центрально-симетричний LBP оператор майже не поступається у ефективності облич класичному і рівномірному LBP операторам.
3. Також зроблений висновок, що оптимальним вибором розбиття зображення на блоки по співвідношенню витрат пам'яті і ефективності розпізнавання являється розбиття 4x4.

4. РОЗРОБКА СИСТЕМИ РОЗПІЗНАВАННЯ ОБЛИЧ

4.1. Інструментарій розробки

Розробка системи велась на об'єктно-орієнтованій мові програмування C# в середовищі розробки Microsoft Visual Studio 2017. C# відноситься до сімейства мов з C-подібним синтаксисом, з яких його синтаксис найбільш близький до C++ та Java. Мова має статичну типізацію, підтримує поліморфізм, перегрузку операторів, делегати, атрибути, властивості, загальні типи та методи, ітератори, анонімні функції з підтримкою замикань, LINQ, виключення, коментарі у форматі XML[27].

Наявність механізму «збір мусора» в C# дозволить ефективно організовувати роботу з списковими структурами, на яких основана робота системи.

З ціллю полегшення процесу розробки було вирішено використовувати бібліотеку OpenCV. Дана бібліотека розроблена на C/C++, а також має інтерфейси для Python, Java і інших мов, в тому числі і обгортку для .NET мов – EmguCV, яка і була використана у роботі. Підтримує Windows, Linux, Mac OS, iOS і Android. Вона містить алгоритми для обробки, реконструкції і очистки зображень, розпізнаванню образів, захвату відео, спостереженням за об'єктами, калібровки камер і інший. Бібліотека розповсюджується по ліцензії BSD, а значить, може вільно користуватись в академічних цілях і комерційним цілям[28][29].

Наявність великої кількості реалізованих алгоритмів комп'ютерного зору, а також великий набір статей теоретичних матеріалів по їх застосуванню роблять бібліотеку OpenCV ідеальним вибором для використання в проектах, присвячених рішенням проблем комп'ютерного зору.

4.2. Аналіз потоків даних

У відповідності з загальним алгоритмом обробки кадрів відеопотоку можна скласти діаграму потоків даних розроблюваної системи. Дана діаграма представлена на (рис. 15).

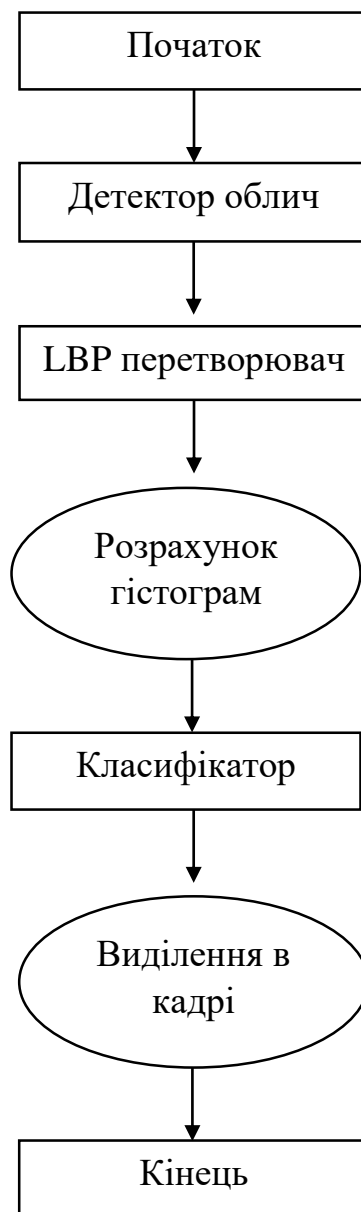


Рисунок 15 – Діаграма потоків даних системи

Виходячи з діаграми потоків даних, доцільно розробити класи, які відповідають сутностям, які беруть участь у перетворенні даних. До таких сутностей можна віднести детектор облич і LBP перетворювач.

Класифікатор має сенс реалізувати методом головного класу додатка, тому що він буде тісно взаємодіяти зі списками розпізнаних і не розпізнаних облич, які є атрибутами головного класу. Окрім цього, необхідно реалізувати функцію розрахунку гістограм, клас, який описує категорію облич, і клас, який описує розпізнане обличчя.

Детектор облич має реалізувати алгоритми, які описані в підрозділі 2.1 даної роботи. На вхід детектора надходять кадри відеопотоку, на яких він шукає особи. В результаті отримуємо список прямокутних областей кадру, відповідних положенням осіб на ньому. Сам каскадний класифікатор доцільно реалізувати засобами бібліотеки Emgu CV, в якій є його реалізація з набором якісно навчених каскадів. Облікові структури будуть описуватися засобами бібліотеки колекцій мови C#.

LBP перетворювач повинен реалізовувати алгоритм перетворення зображень центральньо-симетричним LBP оператором. На вхід перетворювача будуть подаватися зображення облич. Вихідними даними є перетворені LBP зображення.

Клас головної форми додатка буде містити елементи інтерфейсу програми, а також список розпізнаних і список нерозпізнаних осіб, якими буде оперувати система при обробці кадрів. Крім цього даний клас реалізує функції розрахунку гістограм і графічного відображення результатів роботи системи. Детальний опис елементів інтерфейсу і механізмів для користувача взаємодії з ними будуть описані в розділі, присвяченому розробці інтерфейсу системи.

4.3. Опис класів

В даному підрозділі наведений опис розроблених в ході роботи над проектом класів, які відповідають сутностям, представленим в попередньому підрозділі. UML діаграма класів додатка наведена на (рис. 16).

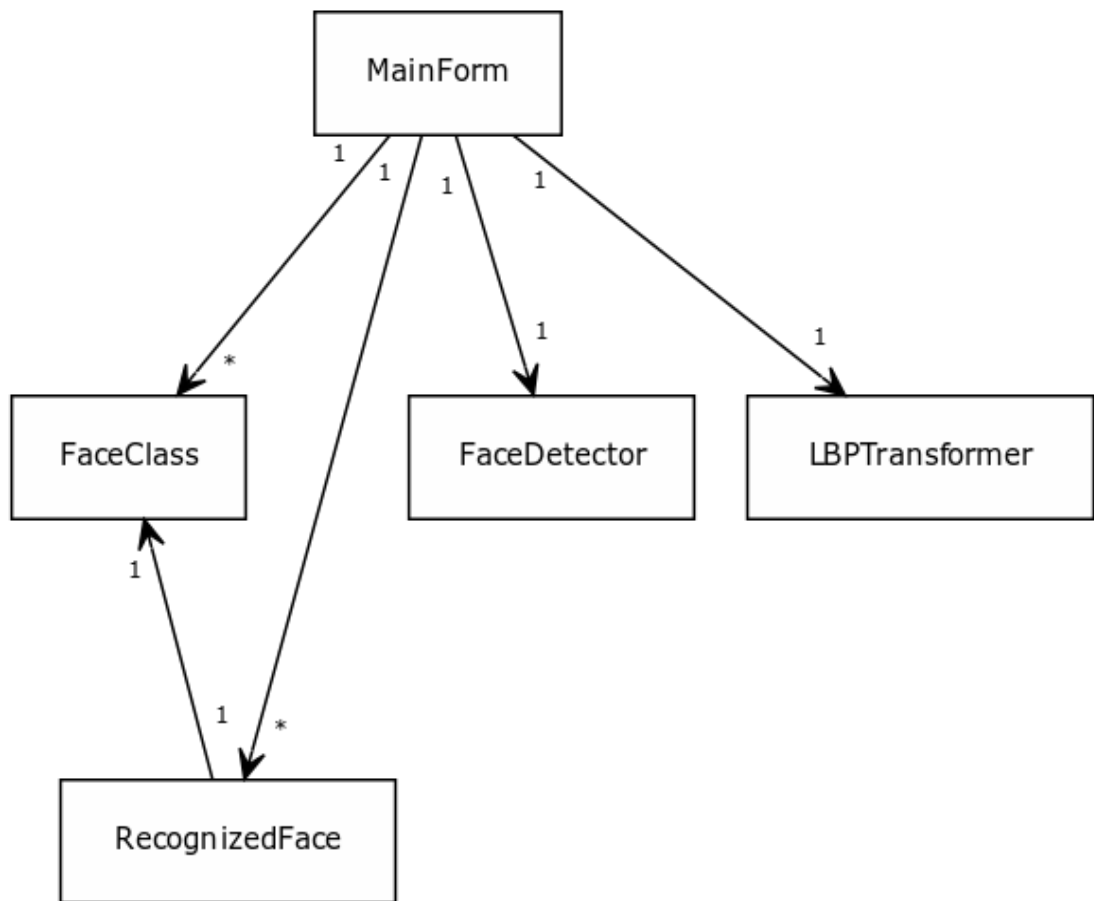


Рисунок 16 – UML діаграма класів застосунка

З міркувань компактності, в представленій діаграмі відображені тільки асоціативні зв'язки класів без їх вмісту. Головний клас додатки MainForm містить список класів осіб типу FaceClass і список розпізнаних осіб типу RecognizedFace. Так само він використовує в своїй роботі детектор осіб FaceDetector і LBP перетворювач зображень LBPTransformer. Кожне розпізнане обличчя RecognizedFace при цьому посилається на відповідний йому клас осіб FaceClass.

Вихідний код представлених класів на мові програмування C # з докладними коментарями представлений в додатку до пояснювальної записки.

4.3.1 RecognizedFace

Клас `RecognizedFace` описує розпізнане класифікатором обличчя. Елементи даного класу зберігаються в оновлюваному списку розпізнаних осіб, на основі даних якого проводиться відображення інформації на формі.

Атрибути класу:

private FaceClass _faceClass - клас осіб, до якого належить дане розпізнане обличчя.

private Rectangle _rect - прямокутна область, яка відповідає положенню обличчя в кадрі відеопотоку.

private Mat _histogram - LBP гістограма особи.

private Image <Gray, Byte> _face - зображення обличчя, взяте з оброблюваного в поточний момент кадру відеопотоку.

private Image <Gray, Byte> _lbp - LBP представлення зображення *_face*.

private double _distance - дистанція між LBP гістограмою обличчя і LBP гістограмою найближчого елемента його класу.

Методи класу:

public RecognizedFace (FaceClass fc, Rectangle r, Image <Gray, Byte> f, Image <Gray, Byte> l, Mat h, double d) - конструктор класу. В якості параметрів передаються значення, що присвоюються атрибутам створюваного об'єкта.

public void SetFaceClass (FaceClass fc) - присвоює значення параметра атрибуту *_faceClass*.

public void SetRect (Rectangle r) - присвоює значення параметра атрибуту *_rect*.

public void SetHist (Mat h) - присвоює значення параметра атрибуту *_histogram*.

public void SetLBP (Image <Gray, Byte> l) - присвоює значення параметра атрибуту *_lbp*.

public void SetFace (Image <Gray, Byte> f) - присвоює значення параметра атрибуту *_face*.

public void SetDist (double d) - присвоює значення параметра атрибуту *_distance*.

public FaceClass GetFaceClass () - повертає значення атрибуту *_faceClass*.

public Rectangle GetRect () - повертає значення атрибуту *_rect*.

public Mat GetHist () - повертає значення атрибуту *_histogram*.

public Image <Gray, Byte> GetLBP () - повертає значення атрибуту *_lbp*.

public Image <Gray, Byte> GetFace () - повертає значення атрибуту *_face*.

public double GetDist () - повертає значення атрибуту *_distance*.

4.3.2 LBPTransformer

Клас *LBPTransformer* описує CS-LBP перетворювач. Даний клас є статичним класом, який не має атрибутів і містить тільки один метод, який виконує центрально-симетричні LBP перетворення зображення. Буде доповнюватися методами LBP перетворення при подальшому розвитку розробки.

Методи класу:

public static Image<Gray, Byte> CSTransform(Image<Gray, Byte> input)

– здійснює LBP перетворення вхідного зображення *input* і повертає перетворене зображення.

4.3.3 FaceDetector

Клас *FaceDetector* описує детектор облич, який використовує в своїй роботі метод Віоли-Джонса.

Атрибути класу:

private CascadeClassifier _haar – каскадний класифікатор, за допомогою якого відбувається виявлення облич.

Методи класу:

public FaceDetector() – конструктор класу. Завантажує в атрибут *_haar* навчений каскад розпізнавання облич у вигляді xml файла

public Rectangle[] GetFacesRect(Image<Bgr, Byte> frame, double scaleFactor, int minNeighbors, int sz) – отримує на вхід зображення *frame*, на якому проводиться пошук обличчя, а так само дані для налаштування параметрів виявлення, а саме фактор збільшення скануючого вікна *scaleFactor*, мінімальна кількість вкладених виявлень *minNeighbors* і мінімальний розмір облич *sz*. Повертає список прямокутників, що відповідають положенням облич на зображенні *frame*.

Варто зазначити, що мінімальна кількість вкладених виявлень безпосередньо впливає на точність виявлення облич. Даний параметр задає необхідне для визнання області зображення обличчям кількість спрацьовувань детектора при різних масштабах скануючого вікна, змінює свої розміри відповідно з фактором збільшення *scaleFactor*, в даній області зображення.

Сам процес виявлення облич здійснюється за допомогою функції *CascadeClassifier.DetectMultiScale* бібліотеки Emgu CV, яка викликається для каскадного класифікатора *_haar*. Ця функція являє собою реалізацію каскадної класифікації з методу Віолі-Джонса. Їй же і передаються описані вище параметри.

4.3.4 FaceClass

Даний клас описує клас(категорію) облич. Кожний клас облич відповідає конкретній людині.

Атрибути класу:

private Image<Gray, Byte> _img – зображення обличчя, яке використовується разом з назвою класу для зручності ідентифікації і розрізнення класів.

private Mat[] _histogram – масив LBP гістограм різних зображень обличчя відповідного класу. Використання не одної, а декількох гістограм, обумовлено особливостям методу найближчого сусіда, тому що при розпізнаванні програма шукає найбільш схожого представника класу з усіх наявних класів.

private String _name – ім'я класу облич.

Методи класу:

public FaceClass(Image<Gray, Byte> faceimg, Mat[] facehist, String facename) – конструктор класу. В якості параметрів передаються значення, які присвоюються атрибутам створюваного об'єкта.

public Mat[] GetHist() – повертає значення атрибуту *_histogram*.

public String GetName() – повертає значення атрибуту *_name*.

public Image<Gray, Byte> GetImg() – повертає значення атрибуту *_img*.

4.3.5 MainWindow

Клас *MainWindow* описує головне вікно додатку, а також реалізує головний модуль додатку, який описується в підрозділі 4.2.

Атрибути класу:

private FaceDetector _detector – детектор облич.

private List<FaceClass> _faces – список класів облич, які зберігаються в пам'яті на даний момент.

private List<RecognizedFace> _recognizedFaces – список розпізнаних облич в поточному кадрі відеопотоку.

private List<Rectangle> _notRecognized – список областей поточного кадру відеопотоку, які містять обличчя, які класифікатор не може віднести до якого-небудь з існуючих класів.

private Image<Gray, Byte> _mask – зображення маски значимих областей.

private Image<Gray, Byte>[] _currentFaces – зображення облич, зняті користувачем для формування нового класу облич.

private int _currentFacesCount – поточне число зображень облич для формування нового класу.

private Rectangle[] _facesRect – список областей поточного кадру відеопотоку, які містять обличчя.

private bool _capturing – прапор, який показує чи виконується в даний момент захват відеопотоку.

private Capture _capture – об'єкт, який здійснює захват відеопотоку.

private String _videoPath – шлях до відеофайлу.

private Mat _frame – поточний кадр відеопотоку.

private Image<Bgr, Byte> _frameImg – поточний кадр відеопотоку, який необхідний для здійснення ряду операцій, неможливих з використанням типу *Mat*.

Методи класу:

public MainForm() – конструктор класу.

public Mat CalcHistogram(Image<Gray, Byte> image) – повертає гістограму зображення *image*.

private void RecognizeFace(Rectangle rect, Image<Gray, Byte> face, Image<Gray, Byte> lbp, Mat histogram, decimal threshold) – метод, який здійснює розпізнавання облич і формування списків розпізнаних і нерозпізнаних облич. В якості параметрів методу передається зображення обличчя *face*, відповідна йому область кадру *rect*, LBP-зображення обличчя *lbp* і його гістограма *histogram*, а також поріг розпізнавання *threshold*. На основі цих даних і списку класів облич виконується класифікація обличчя методом найближчого сусіда. Блок-схема реалізації методу найближчого сусіда у функції *RecognizeFace* наведена на (рис. 17).

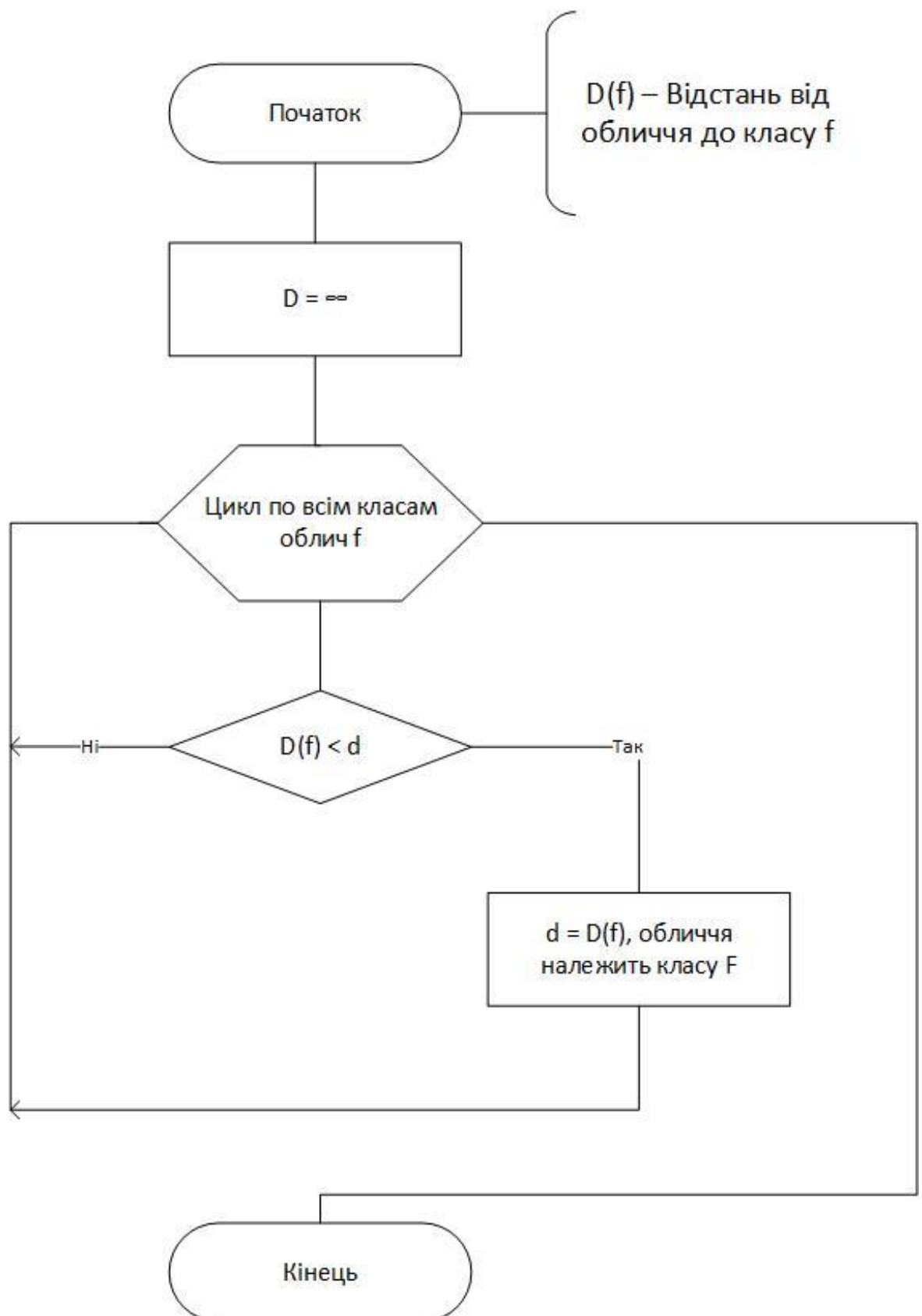


Рисунок 17 – Блок-схема реалізації розпізнавання облич методом найближчого сусіда

Після визначення класу обличчя відбувається порівняння відстані до класу з пороговим значенням, у результаті чого обличчя або переходить до списку розпізнаних облич, або до списку нерозпізнаних облич. Блок-схема алгоритму формування списку розпізнаних облич наведена на (рис. 18).

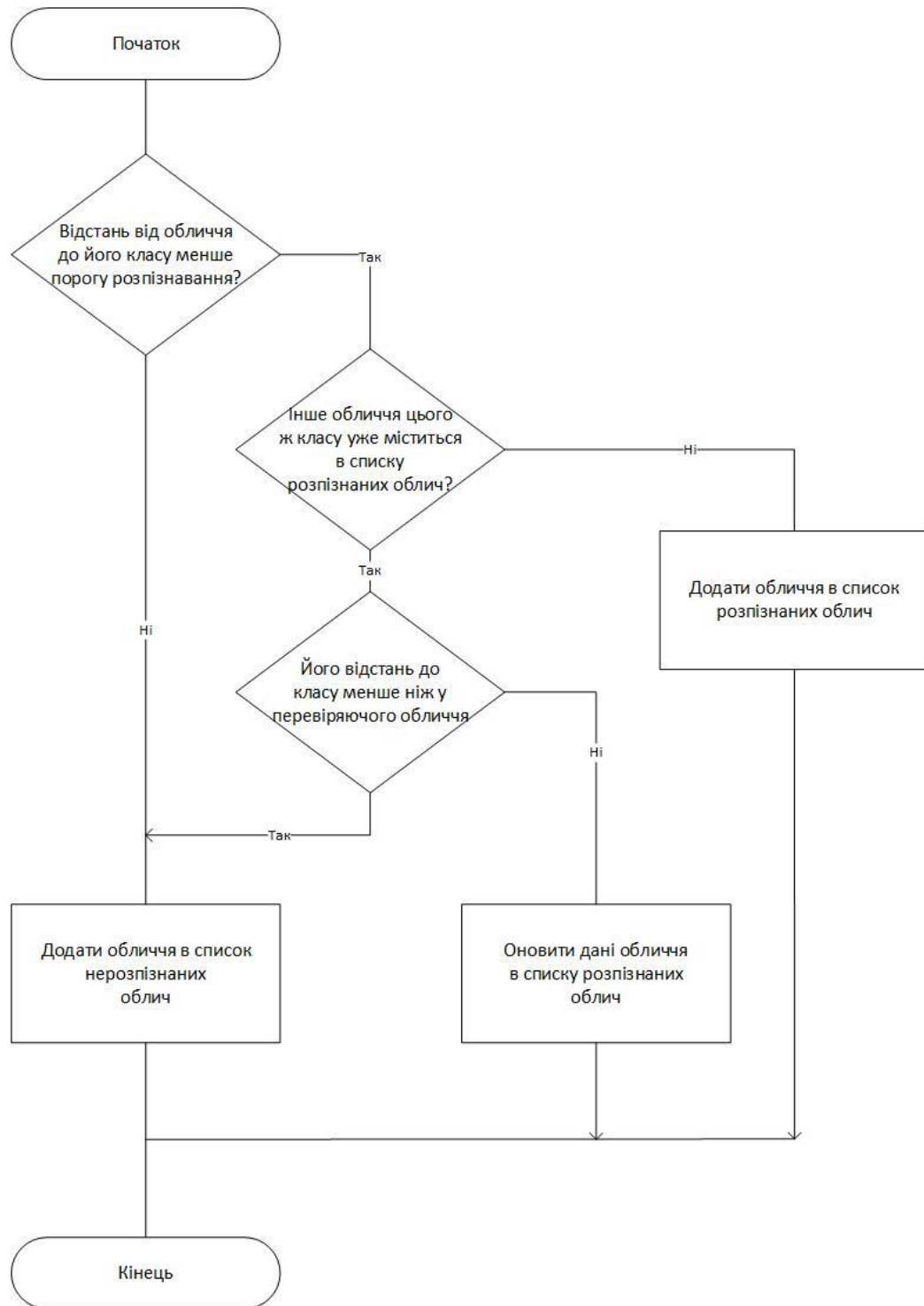


Рисунок 18 – Блок-схема алгоритму формування списків розпізнаних та нерозпізнаних облич

private void ProcessFaces() – даний метод здійснює обробку знайдених облич у кадрі відеопотоку у відповідності з алгоритмом, блок-схема якого наведена на (рис. 19).

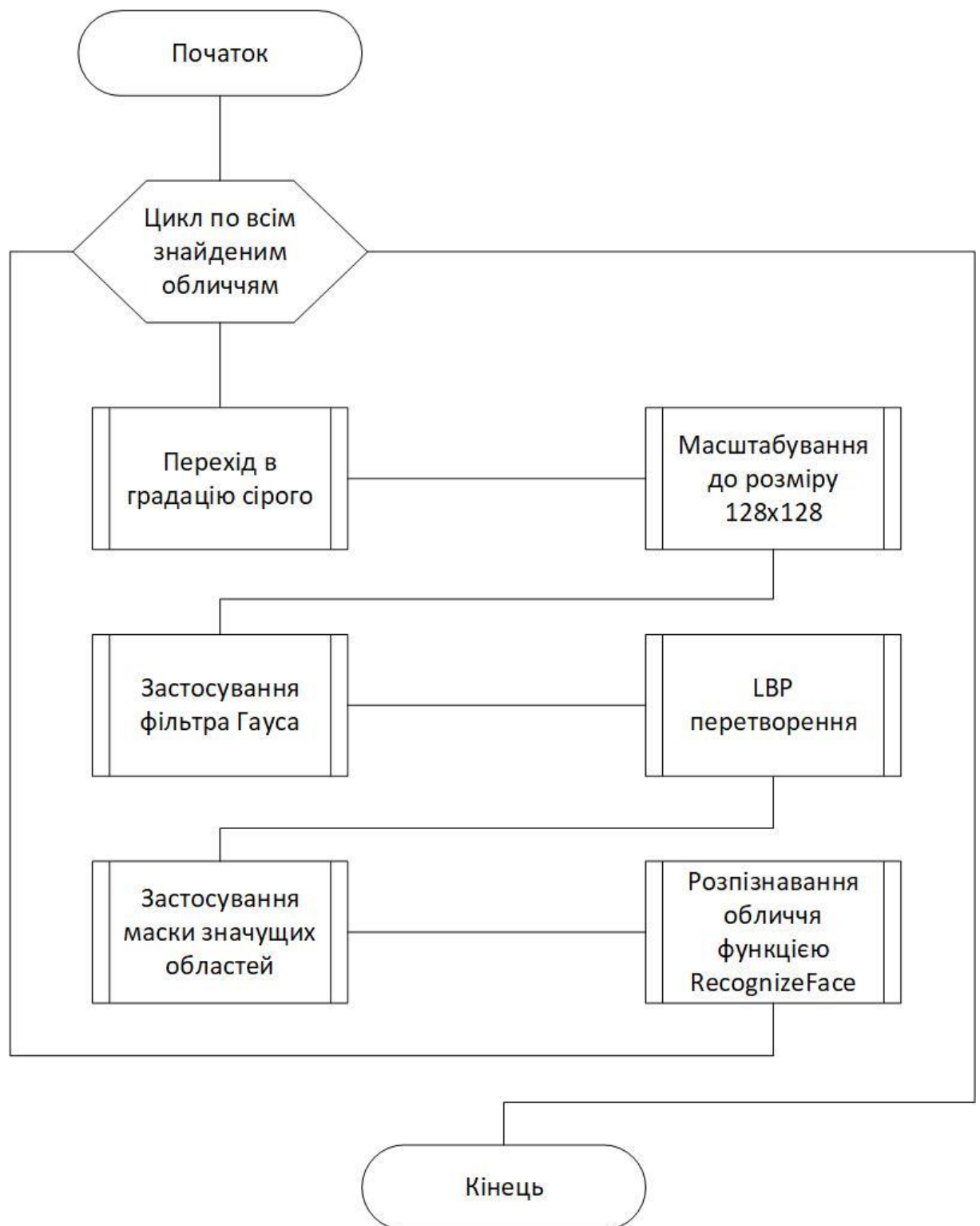


Рисунок 19 – Блок-схема алгоритма обробки обличчя

private void ProcessFrame(object sender, EventArgs arg) – функція обробки кадра відеопотоку. Роботає в соответствии с алгоритмом, блок- схема которого представлена на (рис. 20).



Рисунок 20 – Блок-схема алгоритму обробки кадрів відеопотоку

private void DrawDetected() – дана функція виконує візуальне виділення облич у кадрах відеопотоку і вивід інформації про вибраного користувача лише на форму у відповідності з алгоритмом, представленим на (рис. 21).

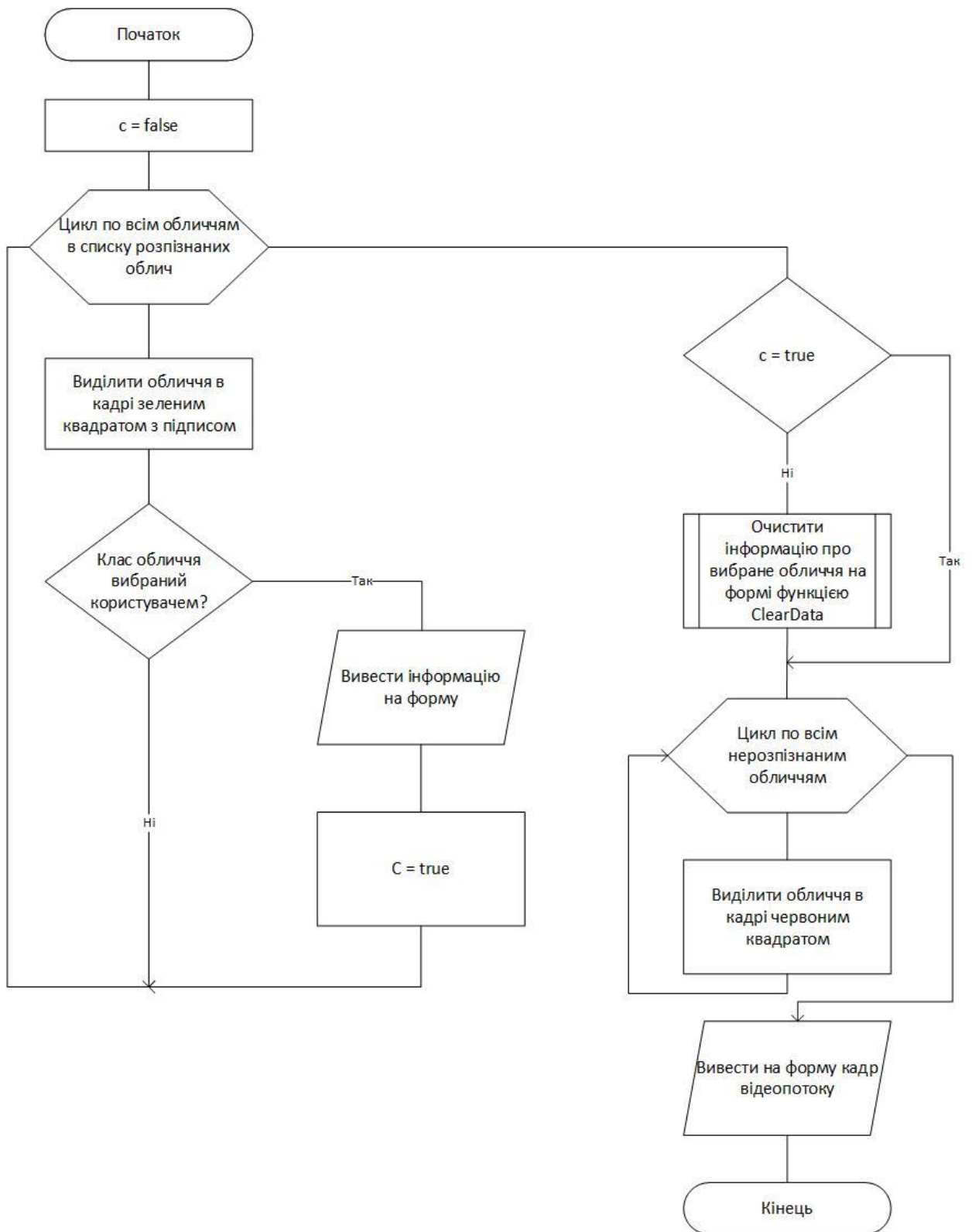


Рисунок 21 – Блок-схема алгоритму виводу інформації про облич на форму

private void ClearData() – метод видаляє інформацію про обраного користувачем обличчя. Використовується кожний раз при оновленні кадру відеопотоку перед завантаженням інформації про обличчя.

private void DrawHistogram(Mat histogram) – метод малює на спеціальному елементі форми гістограму histogram.

private void openFaceClass(object sender, CancelEventArgs e) – метод завантаження у додаток класу обличчя із зовнішнього файлу.

private void addFaceButton_Click(object sender, EventArgs e) – метод, який додає в справжній момент в кадрі обличчя в список зображень, який використовується для формування нового класу облич. Спрацьовує по натисненню кнопки ‘Add Face’ на формі.

private void addClassButton_Click(object sender, EventArgs e) – метод додає до списку класів облич новий клас, створений на основі зображень *_currentFaces*. Спрацьовує по натисненню на кнопку ‘Add Class’ на формі.

private void faceClassesListBox_SelectedIndexChanged(object sender, EventArgs e) – метод, який спрацьовує при зміні обраного класу облич і виводить на форму зображення нового обраного класу.

private void removeFaceButton_Click(object sender, EventArgs e) – метод, який видаляє останнє зображення із списку облич, які використовуються для створення нового класу. Спрацьовує по натисненню на кнопку ‘Remove Face’ на формі.

private void videoButton_Click(object sender, EventArgs e) – метод, який запускає зчитування кадрів з відеопотоку.

private void saveFaceClass(object sender, CancelEventArgs e) – метод, який виконує збереження обраного класу облич у файл. Використовує механізм серіалізації.

private void saveClassButton_Click(object sender, EventArgs e) – метод, який визивається при натисненні на кнопку ‘Save Class’. Відкриває діалогове вікно для збереження файлу.

private void openClassButton_Click(object sender, EventArgs e) – метод, який викликається при натисненні на кнопку ‘Open Class’. Відкриває діалогове вікно для відкриття файлу.

private void removeClassButton_Click(object sender, EventArgs e) – метод, який видаляє обраний клас зі списку. Спрацьовує по натисненню ‘Remove’ на формі.

private void openVideoButton_Click(object sender, EventArgs e) – метод, який викликається при натисненні на кнопку ‘Open File’. Відкриває діалогове вікно вибору відеофайлу.

private void openFileDialog2_FileOk(object sender, CancelEventArgs e) – зберігає шлях до обраного відеофайлу в атрибут *_videoPath*.

4.4 Опис інтерфейсу системи

При розробці інтерфейсу додатку можна обмежитись єдиним головним вікном, яке буде містити в собі активні елементи для налаштування параметрів роботи додатку, і області виводу даних про розпізнані обличчя.

Інтерфейс додатку містить в собі декілька областей: область виводу обробленого відеопотоку з камери, область налаштування параметрів роботи додатку, область роботи з класами і область вивід інформації про обраного користувачем лице. Загальний вигляд форми додаток на (рис. 22).

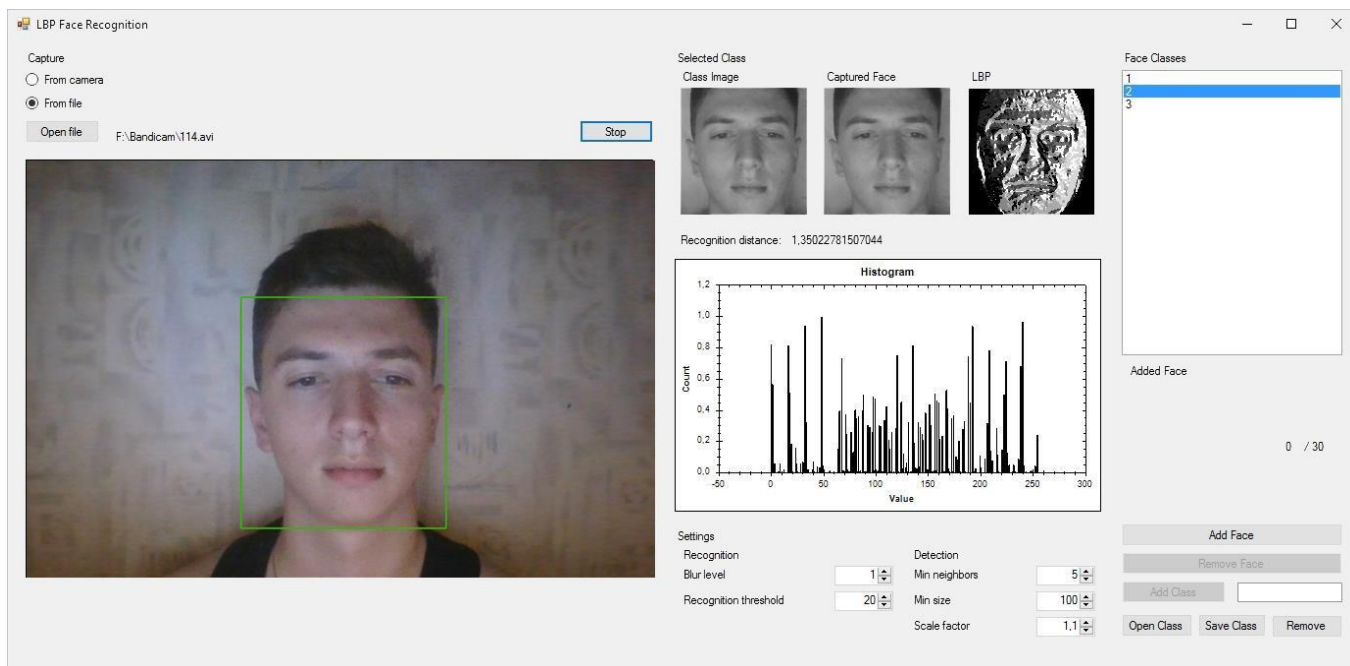


Рисунок 22 – Інтерфейс додатку

Розглянемо детальніше різні області інтерфейсу розробленої системи.

Область виводу відеопотоку призначена для відображення оброблених кадрів. Крім цього, дана область включає кнопки налаштування джерела відеопотоку, кнопки виклику діалогового вікна відкриття відеофайлу, і кнопку, управляючою каналом і зупинкою захвату відеопотоку.

В якості джерела відеопотоку може використовуватись підключена до комп'ютера веб-камера чи відеофайл. Для того, щоб обрати відеофайл-джерело необхідно натиснути кнопку 'Open File'. Після вибору файлу діалоговому вікні і натиснення кнопки 'ОК', шлях до обраного файлу відобразиться на формі. Запуск/зупинка обробки і виводу кадрів з обраного джерела здійснюється шляхом натиснення на кнопки 'Start/Stop'.

Область виводу відеопотоку наведена на (рис. 23).

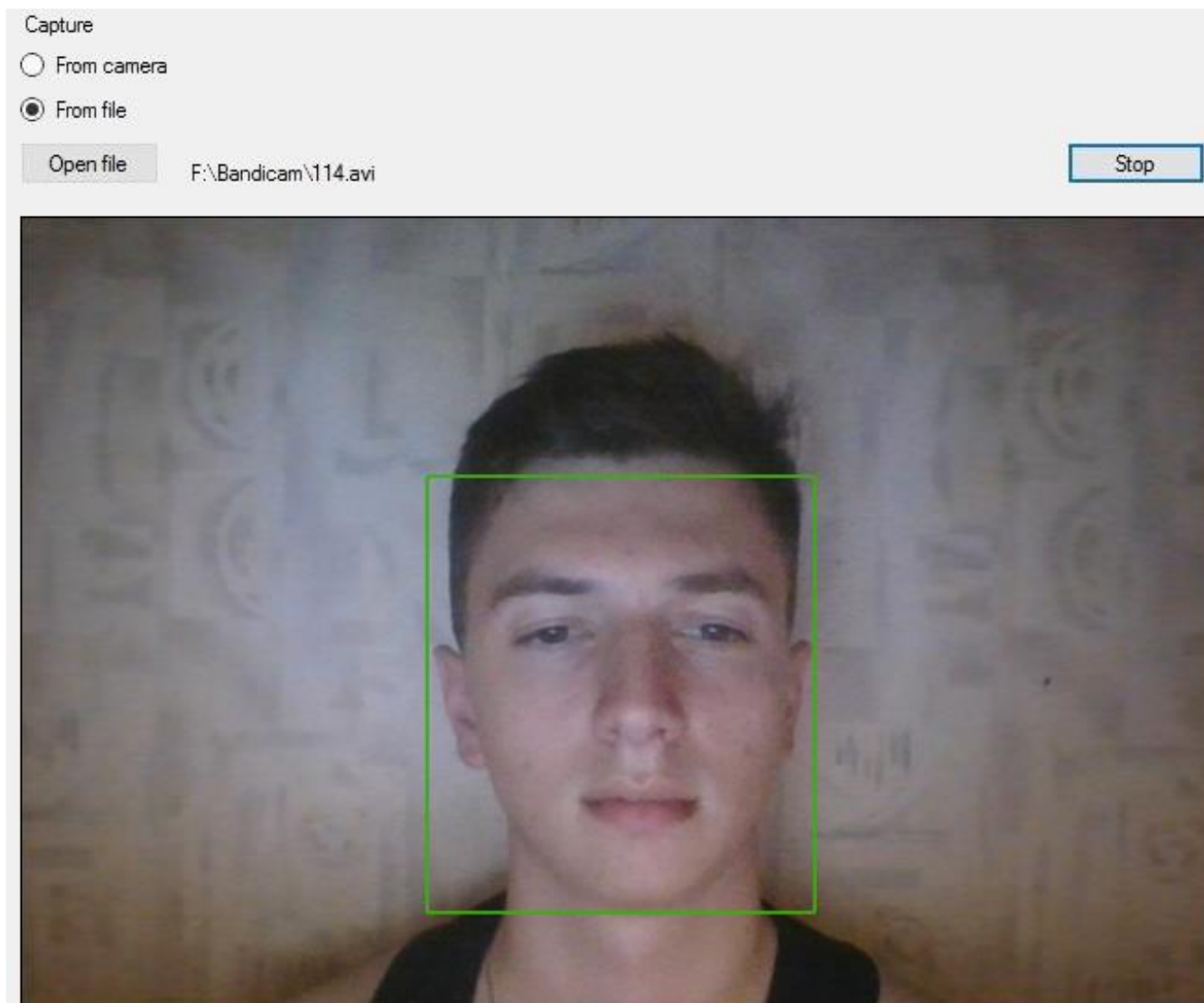


Рисунок 23 – Область виводу відеопотоку

Область виводу інформації про обраного користувачем обличчі призначена для відображення даних розпізнаних додатком облич. Джерелом даних для відображення є обраний користувачем Face Classes клас облич і розпізнане обличчя, яке відповідає цьому класу.

У даній області форми виводяться зображення обраного класу, прямокутна область поточного кадру, яка відповідає обличчю цього класу, а також LBP перетворення зображення обличчя. Крім цього відображається поточний стан між гістограмою розпізнаного обличчя і гістограмою його класу і графічне представлення гістограми LBP зображення обличчя.

Область виводу інформації про обраного користувачем обличчя наведена на

(рис. 24).

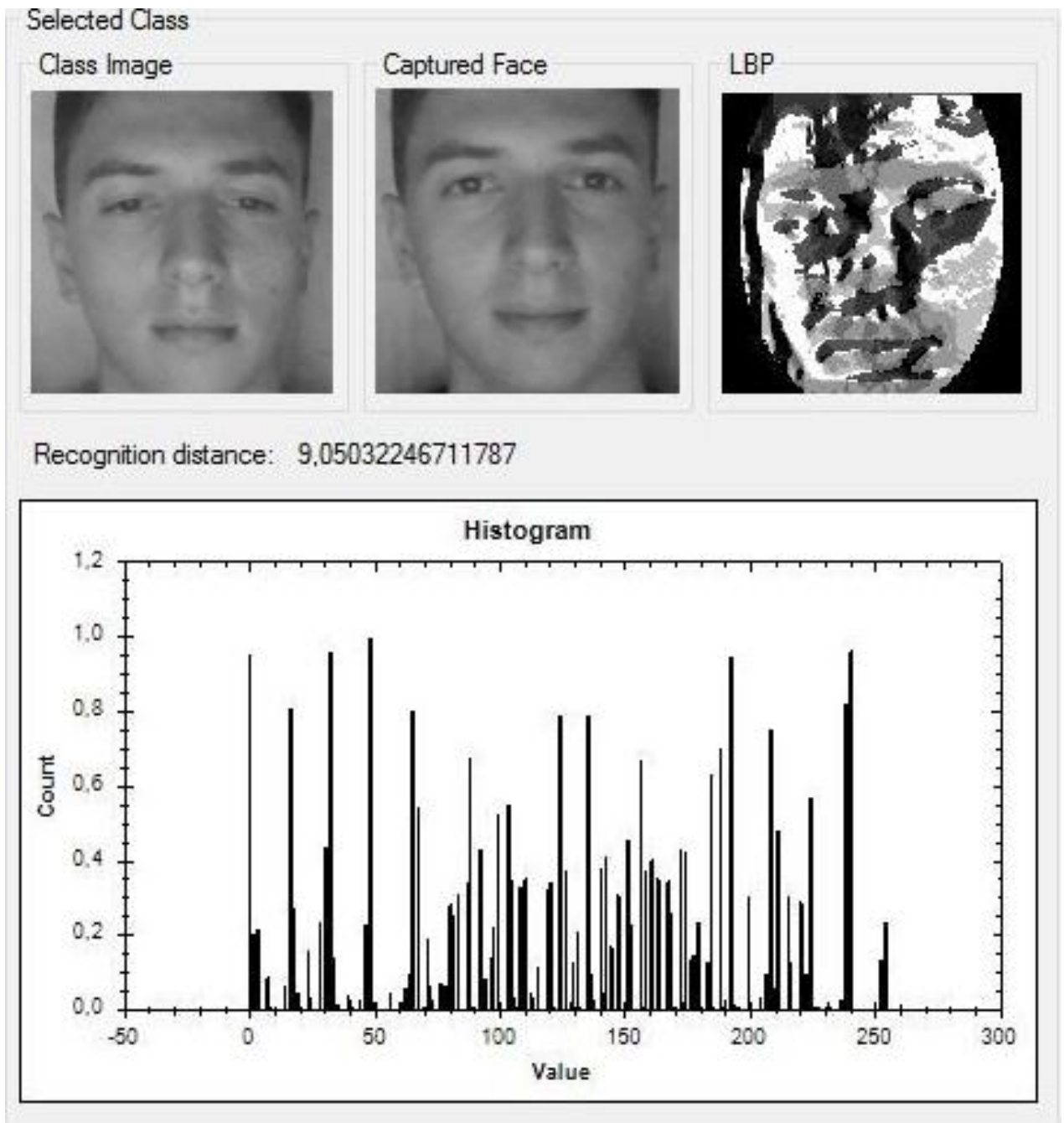


Рисунок 24 – Область виводу інформації про обране обличчя

Область налаштувань додатку включає в себе дві групи налаштувань. Перша група – це налаштування роботи розпізнавання облич. До них відноситься рівень розмиття по Гаусу і поріг розпізнавання. Поріг розпізнавання – максимально допустиме для прийняття рішення про приналежність обличчя класу відстані LBP гістограмою обличчя і гістограмою класу.

Друга група налаштувань даної області містить налаштування детектора обличч. Значення даних налаштувань передаються методу GetFacesRect детектора обличч і були детально описані в попередньому підрозділі.

Область налаштувань роботи додатку наведена на (рис. 25).

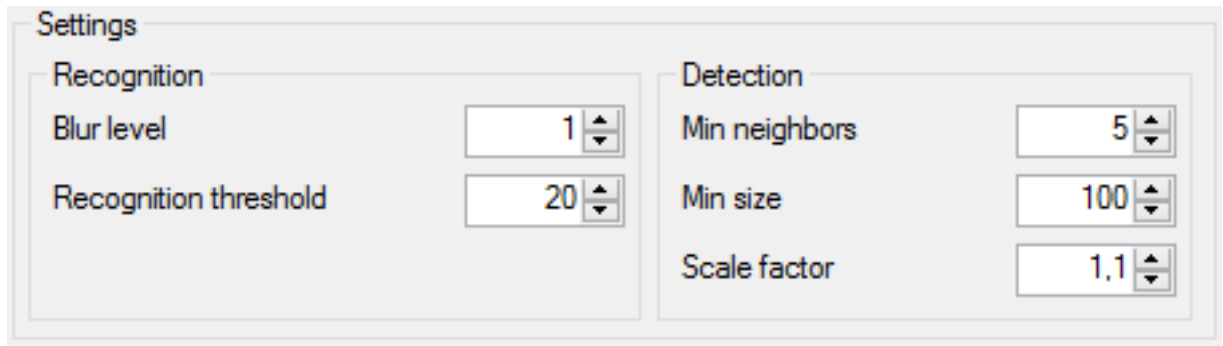


Рисунок 25 – Область налаштувань роботи додатку

Область роботи з класами лиць предназначена для роботи со списком классов лиц, а также формирования новых классов. В верхней части области находится список загруженных в настоящий момент в программу классов лиц. Пользователь может выбирать нужный ему класс нажатием левой клавиши мыши, и информация о выбранном классе и лицах, ему соответствующих, будет отображаться в области вывода информации.

Нижче розміщені елементи інтерфейсу для формування нового класу обличч. Після натиснення на кнопку 'Add Face' в набір зображень для формування класу додається зображення обличчя, яке присутнє в даний момент в кадрі відеопотоку. За допомогою кнопки 'Remove Face' можна видалити останнє зображення з набору. Як тільки необхідний набір зображень (до 30 включно) буде сформований, необхідно ввести ім'я класу в текстове поле і натиснути кнопку 'Add Class' для створення на основі набору зображень нового класу обличч. Створений клас одразу ж відобразиться у списку класів і буде використовуватись при розпізнаванні обличч. Окрім цього в даній області також присутні кнопки для відкриття існуючого класу обличч із файлу, збереження обраного класу у файл та видалення обраного класу зі списку.

Область роботи з класами обличч представлена на (рис. 26).

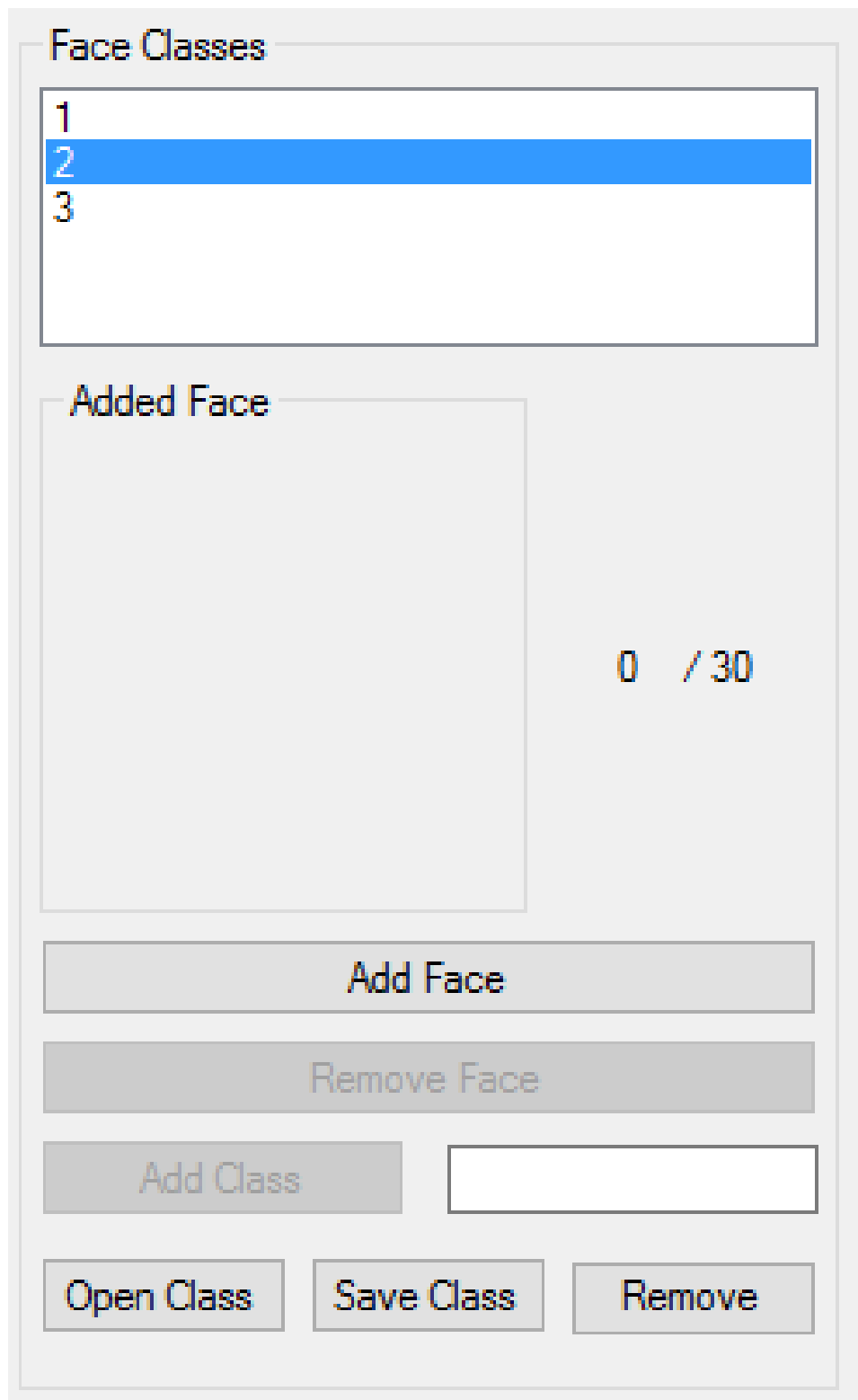


Рисунок 26 – Область роботи з класами облич

Також варто відмітити те, що для запобігання помилок, різні елементи інтерфейсу системи включаються і виключаються в залежності від активності відеопотоку і наявності в ньому зображення облич.

4.5 Тестування розробленої системи

В даному розділі буде описаний процес тестування реалізованого класифікатора облич, а також описано порівняльне дослідження швидкості роботи розробленої системи при використанні різних модифікацій LBP оператора.

4.5.1 Тестування класифікатора облич

Тестування роботи класифікатора облич проводились у відповідності з умовами, описаними у розділі, присвяченому аналізу задачі розпізнавання. Захват відеопотоку проводиться при тих же умовах освітлення, що і зйомка зображень навчальної вибірки.

При цьому використовувались різні положення обличчя при зйомці для збільшення процента вірних класифікацій при переміщенні облич у кадрі.

Всього при тестуванні в програму було завантажено 20 класів облич. Для формування кожного класу використовувалось по 10 зображень. З цих 2 класів облич 15 було створено з використанням зображень людей з мережі інтернет. Дані класи облич були необхідні для того, щоб оцінити якість роботи класифікатора на великому масиві даних, а також, щоб провести тестування роботи програми при читанні відеопотока з файлу.

Приклад роботи розробленої системи розпізнавання облич у відеопотоках представлений на (рис. 27).

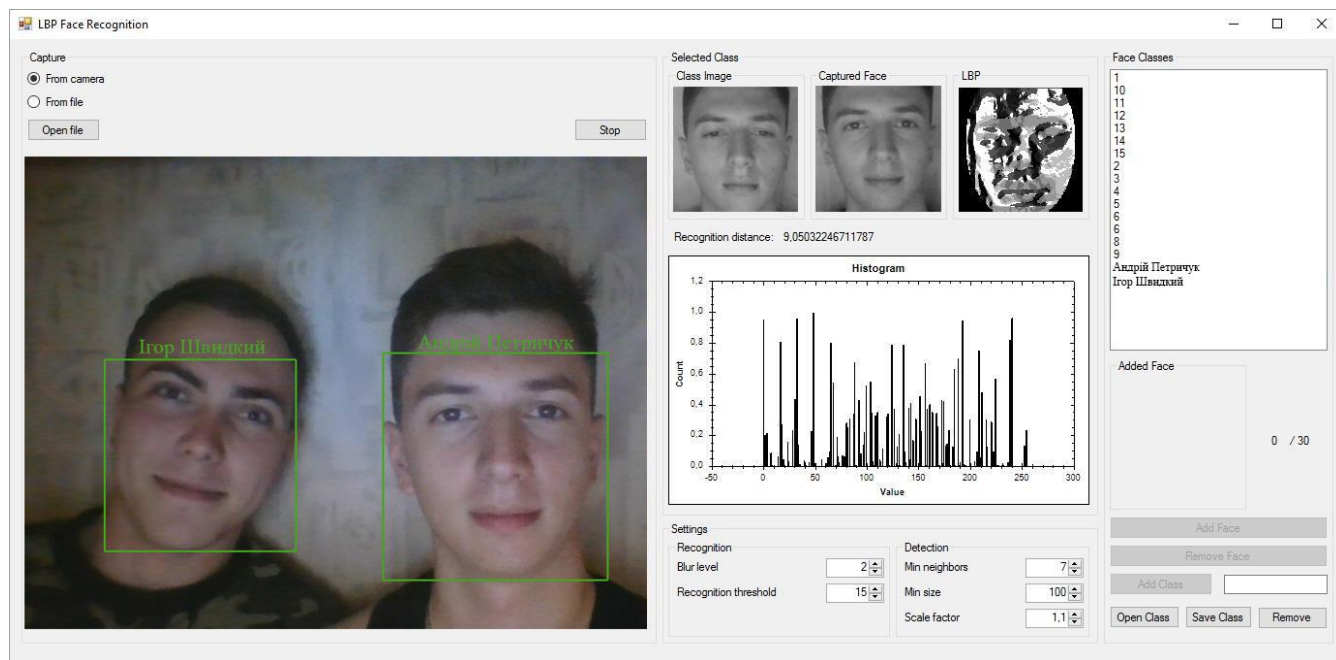


Рисунок 27 – Приклад роботи системи розпізнавання облич

Результати тестування показали, що точність роботи класифікатора при обробці відеопотоку складає більш ніж 90% вірно розпізнаних кадрів, що відповідає результатам, отриманих при порівнянні LBP операторів в розділі 3 даної роботи.

Однак варто відмітити чутливість розробленої системи до сильних немонотонних змін освітлення, а також до змін положення і нахилу розпізнаних облич, не врахованих при зйомці зображень для формування класів облич. При недотриманні необхідних умов правильної роботи додатку, точність класифікації знижується.

4.5.2 Порівняння швидкості роботи LBP операторів

Вибір гістограм центрально-симетричних локальних бінарних шаблонів у якості ознак класифікації був обумовлений високою швидкістю роботи і економією пам'яті при незначних втратах точності розпізнавання в порівнянні з іншими видами LBP операторів. Порівнянню якості їх роботи при класифікації зображень був присвячений 3 розділ даної роботи.

Однак проведені дослідження ніяк не розкриває переваги центрально-

симетричних локальних бінарних шаблонів у швидкості роботи. Для точного визначення переваги у продуктивності і швидкості відеопотоку було проведено порівняння швидкості роботи додатку при використанні різних LBP операторів.

Дослідження проводилось для різних LBP операторів при різній кількості гістограм облич в базі. Швидкість обробки відеопотоку оцінювалась по кількості оброблюваних кадрів в секунду при розпізнавання одного обличчя в кадрі. Результати дослідження наведені в (табл. 4).

Таблиця 4 – тестування швидкості роботи різних LBP операторів

Гістограм в базі Метод	1000	2000	3000	4000	5000	6000	7000	8000	9000	10000
LBP	23	18	15	13	12	10	9	8	8	7
Uniform LBP	30	30	30	29	28	25	23	21	20	19
CS-LBP	30	30	30	30	30	30	30	30	30	30

Тестування проводилось на комп'ютері з процесором Intel Core i5 3 тактовою частотою 3.2 GHz і об'ємом оперативної пам'яті 8 Gb. Розширення кадрів відеопотоку складало 640*480 пікселів. Як видно з результатів тестування, центрально-симетричні локальні бінарні шаблони забезпечують обробку відеопотоку з початковою частотою кадрів 30 кадрів в секунду навіть при наявності 10000 гістограм в базі облич. Інші дві варіації LBP оператора показують значне погіршення продуктивності при подібному збільшенні числа гістограм у базі.

В результаті можна сказати, що центрально-симетричні локальні бінарні шаблони являються кращим вибором серед LBP операторів для рішення задачі розпізнавання облич у відеопотоках у реальному часі.

Висновки

1. Проаналізовані потоки даних розроблюваного додатку. Складена діаграма

потоків додатку.

2. Спроектовані і описані основні модулі розроблюваного додатку.
3. Обраний інструментарій, необхідний для розробки додатку, а саме об'єктно-орієнтована мова програмування c#, бібліотека комп'ютерного зору Emgu CV і середовище розробки Microsoft Visual Studio 2017.
4. Розроблені класи, які відповідають спроектованим модулям і реалізують алгоритм роботи додатку. Приведений детальний опис їх атрибутів і методів.
5. Розроблений графічний інтерфейс користувача додатку. Описані різні робочі області інтерфейсу.
6. Проведено тестування роботи розробленої системи і реалізованого класифікатора облич. Отриманий результат більше 90% вірних розпізнавань облич, при цьому зафіксовано погіршення результатів при недотриманні умов, описаних в розділі, присвяченому аналізу задачі.
7. Проведено порівняння швидкості обробки відеопотоку розроблюваної системи при використанні різних LBP операторів. Результати порівняння показали, що центрально-симетричні локальні бінарні шаблони є кращий вибором серед LBP операторів для вирішення задачі розпізнавання облич у відеопотоках в реальному часі.

ВИСНОВОК

Розроблена система розпізнавання облич у відеопотоках на основі метода Віоли-Джонса і локальних бінарних шаблонів. Для виявлення облич у кадрах відеопотоку був використаний метод Віоли-Джонса. Класифікація виявлених облич виконувалась методом найближчого сусіда з використанням центрально-симетричних локальних бінарних шаблонів.

Тестування розробленої системи показало результати в приблизно 93% вірних розпізнань облич при обробці кадрів відеопотоку з веб-камери в реальному часі, що підтвердило ефективність обраного методу.

Окрім цього в ході роботи було проведено дослідження ефективності різних модифікацій локальних бінарних шаблонів стосовно задачі розпізнавання облич в реальному часі. Було встановлено, що центрально-симетричні шаблони майже не уступають іншим варіантам LBP в якості ознак розпізнавання і при цьому мають вищу швидкість роботи. На основі досліджень зроблено висновок, що гістограми центрально-симетричних локальних бінарних шаблонів є ефективною ознакою для розпізнавання облич в реальному часі.

Розроблена система може використовуватися при вирішенні різних задач відеоаналітики, і, в першу чергу, має безпосереднє застосування в системах контролю доступу і ідентифікації особистості.

Основними напрямленнями подальшого розвитку розробленого методу можна назвати покращення роботи класифікатора облич. Також варто відмітити, що архітектура розробленого додатку дозволяє легко замінювати окремі модулі, що відкривають більше можливості для подальшого розвитку і покращення системи на різних апаратних платформах.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Video Analytics Market worth \$3,971.2 Million by 2020 // Markets and Markets. URL: <http://www.marketsandmarkets.com/PressReleases/iva.asp>.
2. V. Gupta, D. Sharma. A Study of Various Face Detection Methods // International Journal of Advanced Research in Computer and Communication Engineering Vol. 3, May 2014, №5. C. 6694–6697.
3. P. Viola, M. Jones. Rapid object detection using a boosted cascade of simple features / 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Vol. 1. 8–14 December 2001 / The Institute of Electrical and Electronics Engineers, Inc. C. 511–518.
4. Y. Freund, R. E. Schapire. A Short Introduction to Boosting // Journal of Japanese Society for Artificial Intelligence. 1999, №14(5), C. 771-780.
5. D. Roth, The SNoW Learning Architecture // Technical Report UIUCDCS-R-99-2102 . UIUC Computer Science Department, 1999.
6. M. Nilsson, M. Dahl, I. Claesson. The successive mean quantization transform / Proceedings of IEEE Int. Conf.ICASSP 2005, Vol. 4 / The Institute of Electrical and Electronics Engineers Signal Processing Society C. 429 – 432.
7. H. A. Rowley, S. Baluja, .T. Kanade. Neural Network-Based Face Detection // PAMI, January 1998.
8. E. Osuna, R. Freund, F. Girosi. Training support vector machines:an application to face detection // In Proceedings of Computer Vision and pattern Recognition 1997, C. 130-136.
9. Machine learning methods // Graphicon. URL: <http://www.graphicon.ru/oldgr/ru/publications/text/gc2006avezh.pdf>.
10. T. Rawlinson, A. Bhalerao, L. Wang. Principles and Methods for Face Recognition and Face Modelling // Handbook of research on computational forensics, digital crime and investigation: methods and solutions. IGI Global, C. 53-78.
11. M. Turk, A. Pentland. Eigenfaces for recognition. // Cognitive Neuroscience, 1991,

№3(1), С.71–86.

12. P. N. Belhumeur, J. P. Hespanha, D. J. Kriegman. Eigenfaces vs. Fisherfaces Recognition Using Class Specific Linear Projection // IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 19, July 1997, №7 С. 711–720.
13. T. Ojala, M. Pietikäinen, D. Harwood . A Comparative Study of Texture Measures with Classification Based on Feature Distributions // Pattern Recognition, Vol. 29, 1996, С. 51–59.
14. T. Ahonen, A. Hadid, M. Pietikäinen. Face Description with Local Binary Patterns: Application to Face Recognition // IEEE Trans. Pattern Analysis and Machine Intelligence, 1996, №28(12), С. 2037–2041.
15. G.J. Edwards, T.F. Cootes, C.J. Taylor. Face Recognition Using Active Appearance Models // Computer Vision — ECCV’98, Volume 1407 of the series Lecture Notes in Computer Science. С. 581–595
16. U. Prabhu, K. Seshardi. Facial Recognition Using Active Shape Models, Local Patches and Support Vector Machines // ECE Department Carnegie Mellon University.
17. W.Y. Zhao, R. Chellappa. Image-based Face Recognition — Issues and Methods // Image recognition and Classification, 2002, С. 375–402.
18. Р. Гонсалес, Р. Вудс. Цифровая обработка изображений, М.: Техносфера, 2005. – 1072 с.
19. Местецкий Л.М. Математические методы распознавания образов, М.: МГУ, ВМиК, 2002–2004. – 85 с.
20. O. Papageorgiou, P. Papageorgiou. A general framework for object detection / International Conference on Computer Vision, 1998 / The Institute of Electrical and Electronics Engineers, Inc. С. 555–562.
21. L. G. Shapiro, G. C. Stockman. Computer Vision. Prentice Hall, 2001. – 608 с.
22. M. Heikkilä, M. Pietikäinen. A texture-based method for modeling the background and detecting moving objects // IEEE Transactions on Pattern Analysis and Machine

- Intelligence, 2006, №28(4), С. 657-662.
23. M. Heikkil, M. Pietikainen, C. Schmid. Description of Interest Regions with Center-Symmetric Local Binary Patterns // ICVGIP 2006, С. 58–69.
24. Броневи́ч А. Н. Лекции по методам машинного обучения // URL:
http://window.edu.ru/resource/800/73800/files/lect_Lepskiy_Bronevich_pass.pdf.
25. Cambrige Face Database // Cambrige university. URL:
<http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html>
26. Yale Face Database B // UCSD Computer Vision. URL:
<http://vision.ucsd.edu/~leekc/ExtYaleDatabase/ExtYaleB.html>.
27. Язык программирования С#. Классика Computers Science. 4-е изд. / А. Хейлсбер, М. Торгерсен, С. Вилтамут, П. Голд. СПб.: Питер, 2011, 784 с.
28. About OpenCV // URL: <http://opencv.org/about.html>.
29. EmguCV // URL: http://www.emgu.com/wiki/index.php/Main_Page
30. Вежневeц В. Обнаружение и локализация лица на изображении / В. Вежневeц, А. Дегтярева. // Компьютерная графика и мультимедиа. – 2003. – №1(3).
31. K.-R. Muller. S. Mika, G. Ratsch, K. Tsuda. and B. Scholkopf. An introduction to kernel-based learning algorithms, IEEE Transactions on Neural Networks, 12(2). pp. 181–201, 2001.
32. Jochen Maydt and Rainer Lienhart. Face Detection with Support Vector Machines and a Very Large Set of Linear Features, IEEE ICME 2002, Lousanne, Switzerland, Aug. 2002.
33. Zhang, G., Huang, X., Li, S.Z., Wang, Y., Wu, X.: Boosting local binary pattern (LBP)-based face recognition. In: Li, S.Z., Lai, J.-H., Tan, T., Feng, G.-C., Wang, Y. (eds.) SINOBIOMETRICS 2004. LNCS, vol. 3338, pp. 180– 187. Springer, Heidelberg (2004).
34. Ю. Лифшиц, Методи розпізнавання облич, електронний ресурс, режим доступу: <http://yury.name/modern/08modernnote.pdf>
35. Viola P., Jones M. Rapid object detection using a boosted cascade of simple features // Proc. Int. Conf. on Computer Vision and Pattern Recognition.

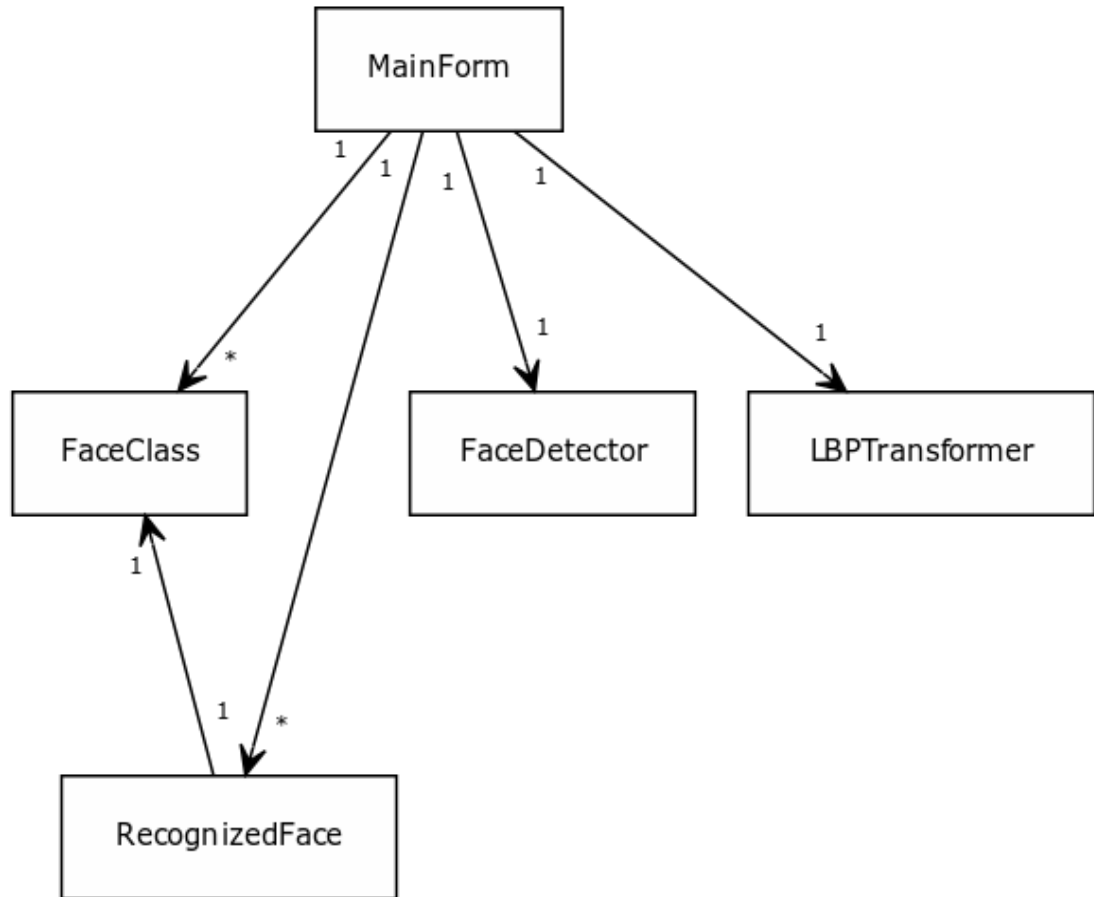
36. Fasel B., Luetttin J. Automatic facial expression analysis: A survey // Pattern Recognition Letters. 2003. V. 36. № 1. P. 259-275.

ДОДАТОК А. КОПІЇ ГРАФІЧНИХ МАТЕРІАЛІВ

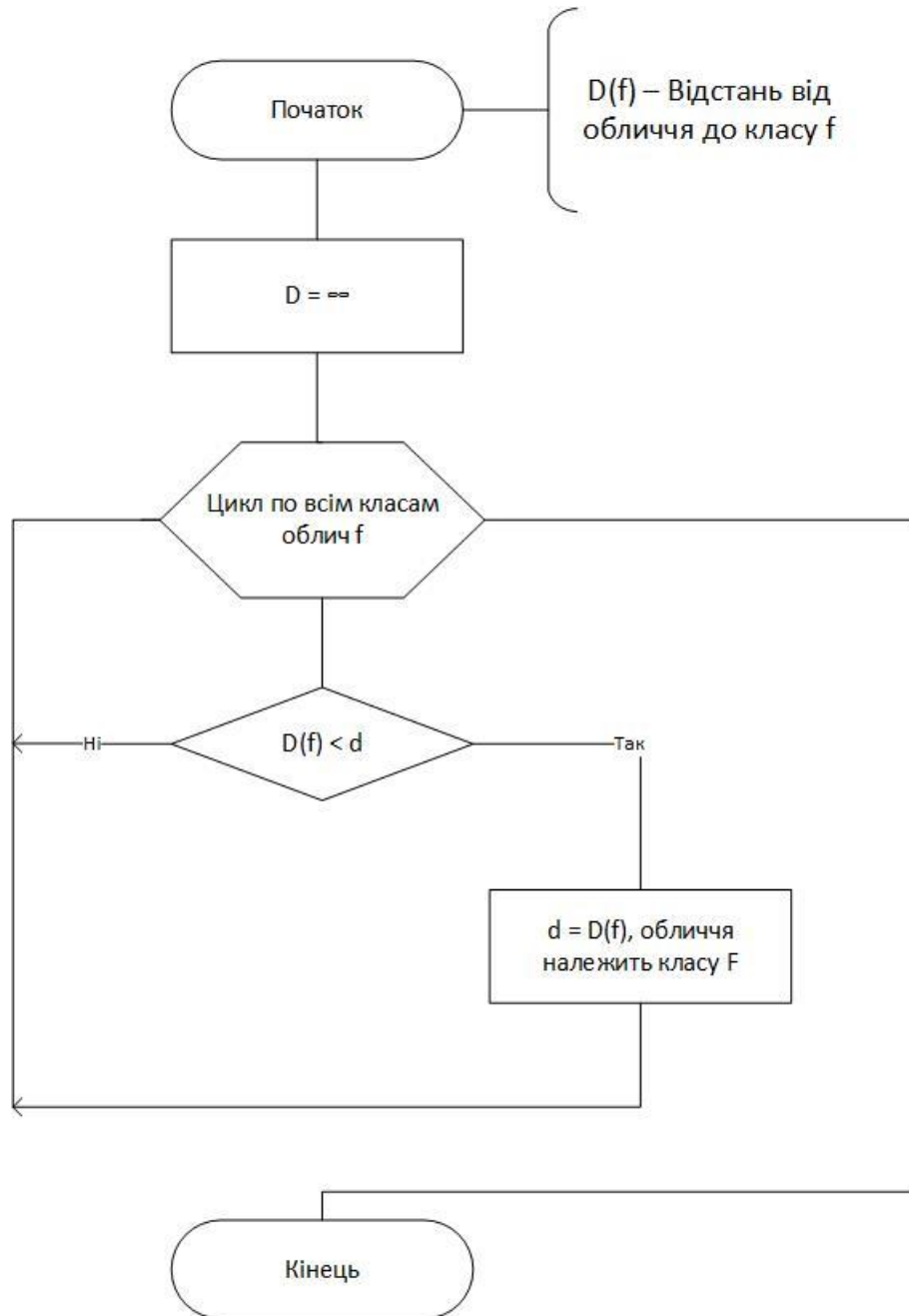
Узагальнена блок-схема алгоритму обробки кадрів відеопотоку



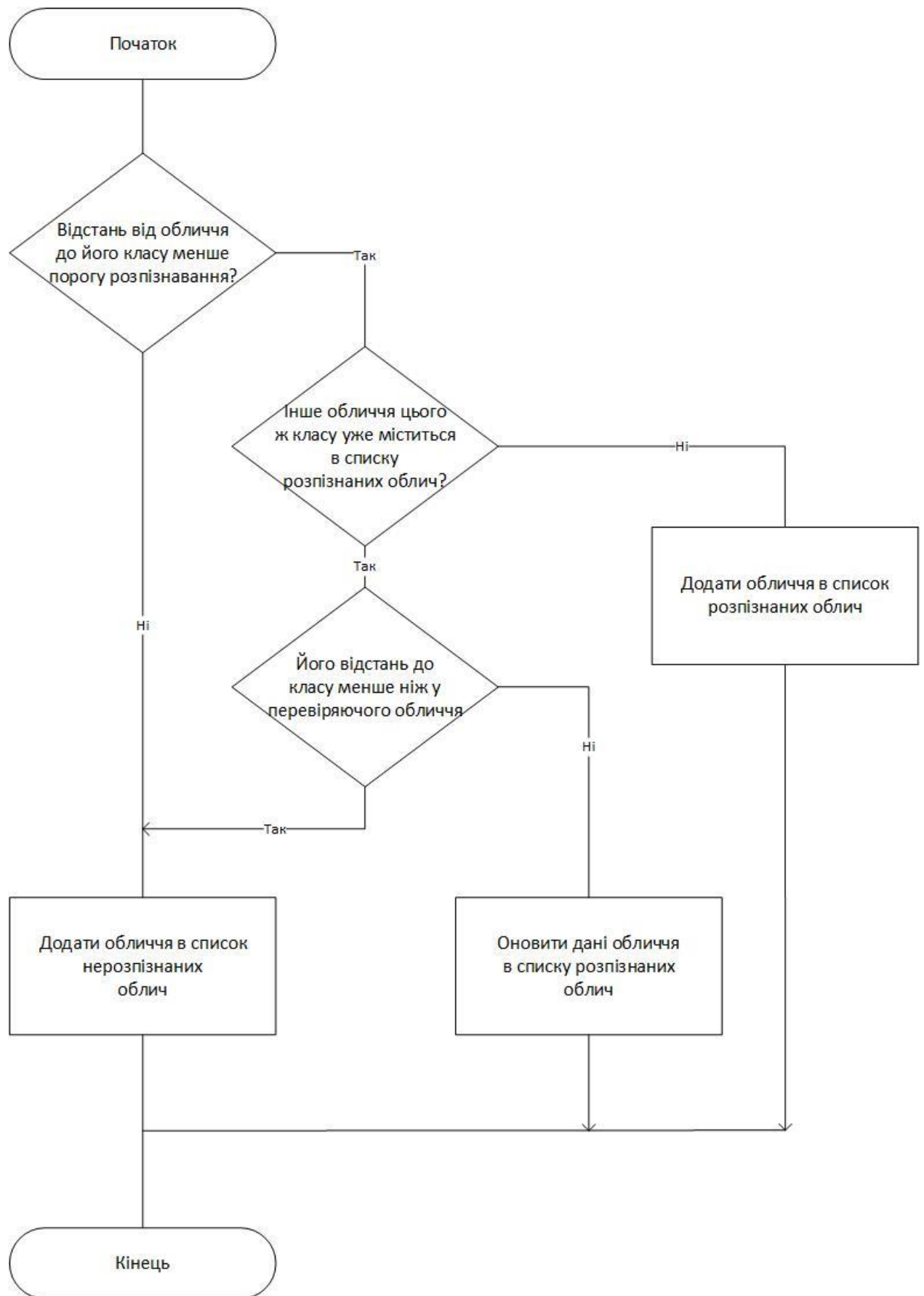
UML-діаграма класів додатку



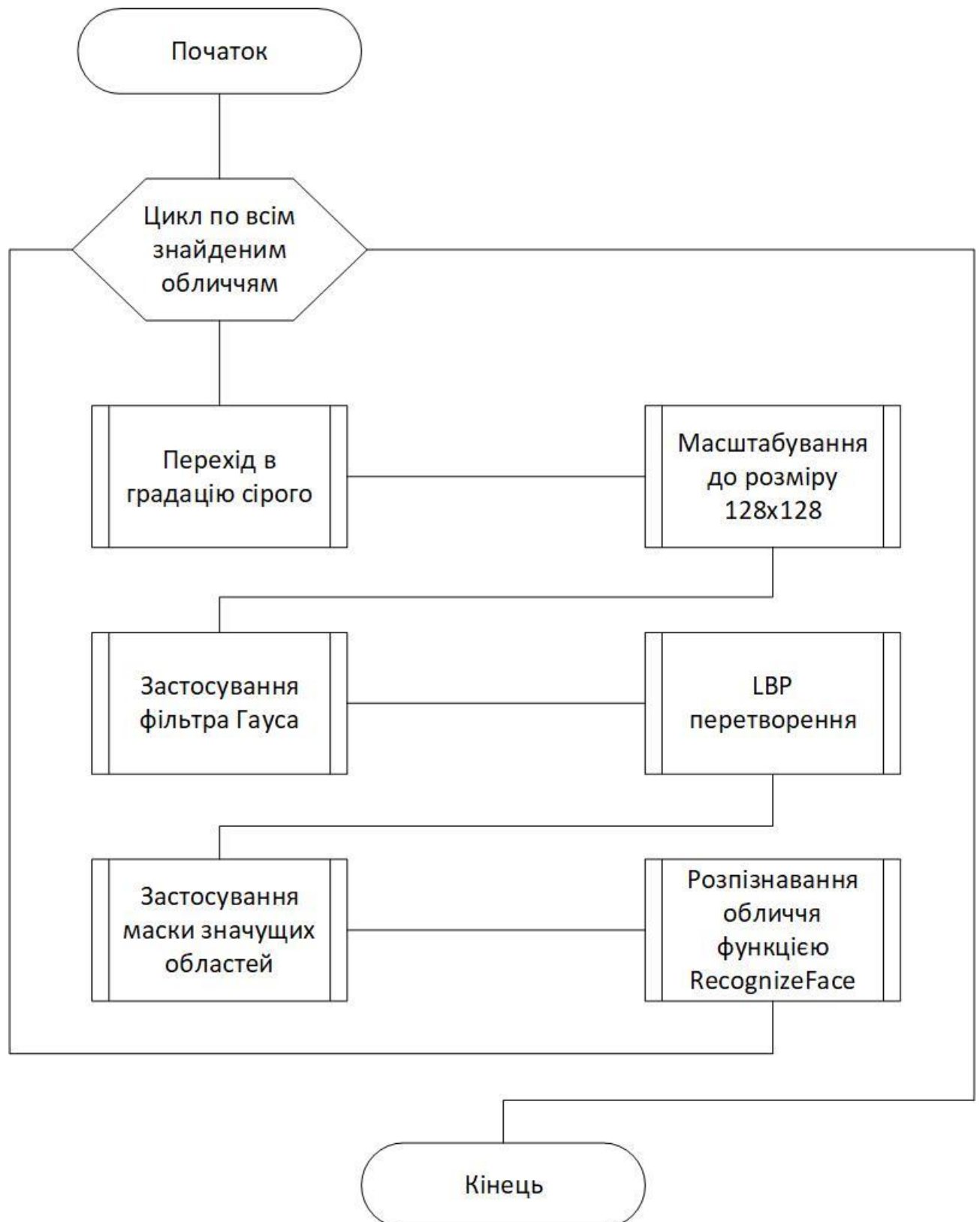
Блок-схема реалізації розпізнавання облич методом найближчого сусіда



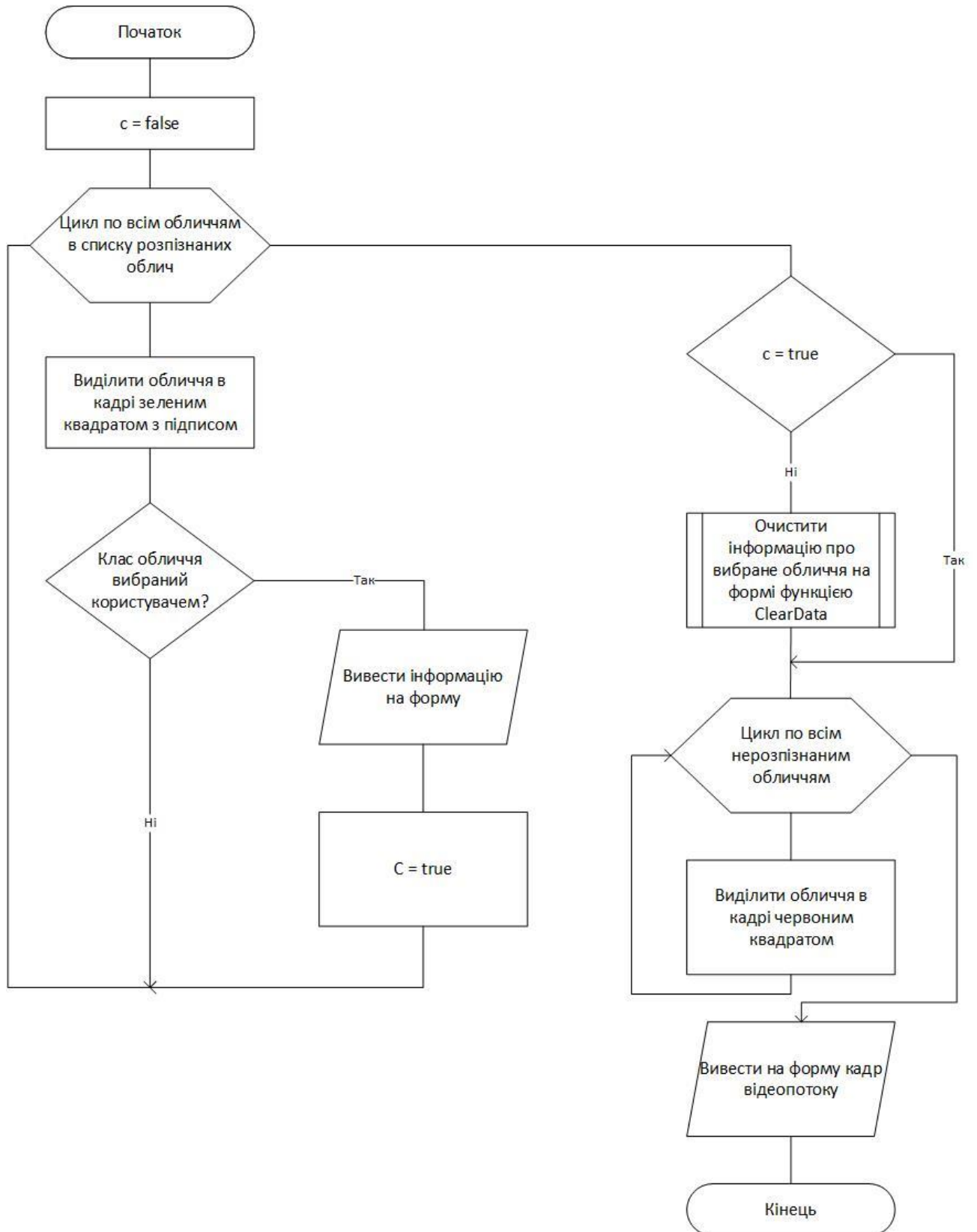
Блок-схема алгоритму формування списків розпізнаних та нерозпізнаних облич



Блок-схема алгоритму обробки виявлених облич



Блок-схема алгоритму інформації про обличчя на форму



ДОДАТОК Б. КОПІЇ ПУБЛІКАЦІЙ ЗА ТЕМОЮ МАГІСТЕРСЬКОЇ ДИСТЕРТАЦІЇ

УДК 004.93'1

Д.т.н., професор Зайцев В.Г., студент Петричук А.В.

**Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»**

СИСТЕМА РОЗПІЗНАВАННЯ ОБЛИЧ У ВІДЕОПОТОКАХ НА ОСНОВІ МЕТОДА ВІОЛІ-ДЖОНСА І ЛОКАЛЬНИХ БІНАРНИХ ШАБЛОНІВ

Abstract

Volodymyr H. Zaitsev, assoc. prof.; Andrii Petrychuk, student

***System of face recognition in video Viola-Jones basis and the local binary templates**
This paper concerns the description of the proposed algorithm for face recognition by using the method of Viola-Jones for detection and nearest neighbor method and histograms of centrally symmetric local binary patterns as signs of classification.*

Вступ

В останній час широке розповсюдження отримує відеоаналітика - технологія, яка використовує комп'ютерний зір для автоматизованого збору різної інформації на основі послідовності кадрів, які отримуються з відеокамер в реальному часі чи з відеозаписів. Дана технологія може бути застосована у відеоспостереженні, системах безпеки, транспорті.

Одна із задач, які вирішуються відеоаналітикою, є розпізнавання облич у відеопотоках. Вирішення цієї задачі в першу чергу має безпосереднє використання в системах контролю доступу і ідентифікації особистості.

Постановка задачі

Метою дослідження є розробка ефективного способу розпізнавання обличчя людини на основі методів Віолі-Джонса та локальних бінарних шаблонів, центрально-симетричних локальних бінарних шаблонів.

Термінологія

Відеоаналітика - технологія, яка використовує комп'ютерний зір для автоматизованого збору різної інформації на основі послідовності кадрів, які отримуються з відеокамер в реальному часі чи з відеозаписів.

LBP (англ. *Local Binary Patterns*) – простий оператор, який використовується для класифікації текстур в комп'ютерному зорі. Він представляє собою опис околиці пікселя зображення у двійковій формі.

Опис архітектури системи розпізнавання облич

Задачу розпізнавання облич на кадрах відеопотоку можна умовно поділити на два етапи. Перший етап – виявлення облич в кадрі. Другий етап – безпосереднє розпізнавання знайдених облич.

Після порівняння ефективності методів виявлення облич, найкращим по показникам відсотка вірних виявлень і помилкам другого роду являється алгоритм AdaBoost[2]. Тому, при створення системи розпізнавання облич у відеопотоку використовувався метод Віюлі-Джонса[3], оснований на даному алгоритмі. Окрім переваг в точності виявлення, даний метод має високу швидкість роботи, що робить його найбільш підходящим для виявлення облич в реальному часі.

Для безпосереднього розпізнавання облич був обраний алгоритм бінарних шаблонів. Переваги цього алгоритму в простоті реалізації і високій швидкості роботи, яку можна збільшити модифікувавши алгоритм, при цьому алгоритм показує високі результати при розпізнавання облич і стійкий до монотонних змін освітлення. Виходячи з цього він найкраще підходить до розпізнавання обличчя в системах обробки в реальному часі.

Окрім етапів виявлення і розпізнавання доцільно використовувати проміжні етапи обробки знайдених облич. Використання фільтрів Гауса після виявлення облич допоможе знизити виявлення шумів при розпізнаванні. Також потрібно застосувати маску значимих областей до локалізованих і перетворених оператором LBP зображень облич[4], яка дозволить зменшити вплив при розпізнавання кутових областей зображення, які містять задній план.

В результаті загальний алгоритм розпізнавання має складатися з наступних кроків: виявлення облич в кадрі, обробка знайдених облич фільтром Гауса, застосування LBP трансформації до знайдених облич аналогічно представлених в [1] з наступним застосуванням маски значимих областей, розрахунок гістограм знайдених облич, класифікація облич по гістограмам методом найближчого сусіда[1]. В результаті буде отриманий список відслідковуваних облич з їх характеристиками і координатами прямокутних областей кадру, в яких вони знаходяться.

Загальна блок-схема алгоритму обробки кадрів розроблюваної системи представлена на рис. 1.



Рис. 1. Узагальнена блок-схема алгоритму обробки кадрів відеопотоку

З точки зору швидкодії ідеальним вибором в якості ознак класифікації при розробці системи розпізнавання в реальному часі являються центральні-симетричні локальні бінарні шаблони[5]. В результаті проведених досліджень LBP операторів з

використанням бази зображень облич Кембриджського та Єльського університетів було зроблено висновок, що для досягнення точності $>90\%$ доцільно використовувати розбиття зображень від 4×4 і більше. Розбиття на 4×4 і більше - оптимальне рішення по співвідношенню точності і витрат пам'яті при розбитті зображення на області. Це розбиття забезпечує стабільно високий відсоток правильних класифікацій при не дуже великих витратах пам'яті.

Висновки

У статті запропоновано спосіб для розпізнавання облич у відеопотоках на основі метода Віюлі-Джонса і локальних бінарних шаблонів. Для виявлення облич в кадрах відеопотоку був запропонований метод Віюлі-Джонса. Класифікація виявлених облич виконувалась методом найближчого сусіда з використанням центрально-симетричних локальних бінарних шаблонів.

Тестування розробленої системи виявило результати в приблизно 94% правильних розпізнаних облич при обробці кадрів з відеопотоку з веб-камери реального часу.

Окрім цього було проведено дослідження ефективності різних модифікацій локальних бінарних шаблонів, які застосовувались для розпізнавання облич в реальному часі. Було встановлено, що центрально-симетричні локальні бінарні шаблони можуть бути використані в якості ознак розпізнавання і при цьому мають більш високу швидкість роботи.

Також варто відмітити, запропонована архітектура системи розпізнавання облич дозволяє легко замінювати окремі модулі, що відкриває більше можливості для подальшого розвитку, модифікації та покращення системи.

Література

1. *Ahonen, A. Hadid, M. Pietikainen.* Face Description with Local Binary Patterns: Application to Face Recognition // IEEE Trans. Pattern Analysis and Machine Intelligence, 1996, №28(12), С. 2037-2041.
2. *H. A. Rowley, S. Baluja, T. Kanade.* Neural Network-Based Detection // PAMI, January 1998.
3. *P. Viola, M. Jones.* Rapid object detection using a boosted cascade of simple features / 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognitions Vol. 1. 8-14 December 2001 / The institute of Electrical and Electronics Engineers, Inc. С. 511-518.
4. *M. Heikkila, M. Pietikainen. C. Schmid.* Description of Interest Regions with Center-Symmetric Local Binary Patterns // ICVGIP 2006, С.58-69.
5. *M. Heikkila, M. Pietikainen.* A texture-based for modeling the background and detecting moving objects // IEEE Transactions on Pattern Analysis and Machine Intelligence, 2006, №28(4), С. 657-662.

Зайцев В.Г., Петричук А.В.

*Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»,
andrei.petrychuk@gmail.com*

**Розпізнавання облич у відеопотоках на основі метода віюлі-джонса і
локальних бінарних шаблонів
Abstract**

Вступ

В останній час широке розповсюдження отримує відеоаналітика - технологія, яка використовує комп'ютерний зір для автоматизованого збору різної інформації на основі послідовності кадрів, які отримуються з відеокамер в реальному часі чи з відеозаписів. Дана технологія може бути застосована у відеоспостереженні, системах безпеки, транспорті.

Одна із задач, які вирішуються відеоаналітикою, є розпізнавання облич у відеопотоках. Вирішення цієї задачі в першу чергу має безпосереднє використання в системах контролю доступу і ідентифікації особистості.

Постановка задачі

Метою дослідження є розробка ефективного способу розпізнавання обличчя людини на основі методів Віюлі-Джонса та локальних бінарних шаблонів, центрально-симетричних локальних бінарних шаблонів.

Опис архітектури системи розпізнавання облич

Задачу розпізнавання облич на кадрах відеопотоку можна умовно поділити на два етапи. Перший етап – виявлення облич в кадрі. Другий етап – безпосереднє розпізнавання знайдених облич.

Після порівняння ефективності методів виявлення облич, найкращим по показникам відсотка вірних виявлень і помилкам другого роду являється алгоритм AdaBoost[1]. Тому, при створення системи розпізнавання облич у відеопотоку використовувався метод Віюлі-Джонса[2], оснований на даному алгоритмі. Окрім переваг в точності виявлення, даний метод має високу швидкість роботи, що робить його найбільш підходящим для виявлення облич в реальному часі.

Для безпосереднього розпізнавання облич був обраний алгоритм бінарних шаблонів. Переваги цього алгоритму в простоті реалізації і високій швидкості роботи, яку можна збільшити модифікувавши алгоритм, при цьому алгоритм показує високі результати при розпізнавання облич і стійкий до монотонних змін освітлення. Виходячи з цього він найкраще підходить до розпізнавання обличчя в системах обробки в реальному часі.

Окрім етапів виявлення і розпізнавання доцільно використовувати проміжні етапи обробки знайдених облич. Використання фільтрів Гауса після виявлення облич допоможе знизити виявлення шумів при розпізнаванні. Також потрібно застосувати маску значимих областей до локалізованих і перетворених оператором LBP зображень облич[3], яка дозволить зменшити вплив при розпізнаванні кутових областей зображення, які містять задній план.

В результаті загальний алгоритм розпізнавання має складатися з наступних кроків: виявлення облич в кадрі, обробка знайдених облич фільтром Гауса, застосування LBP трансформації до знайдених облич аналогічно представлених в [1] з наступним застосуванням маски значимих областей, розрахунок гістограм знайдених облич, класифікація облич по гістограмам методом найближчого сусіда[4]. В результаті буде отриманий список відслідковуваних облич з їх характеристиками і координатами прямокутних областей кадру, в яких вони знаходяться.

Алгоритм обробки кадрів складається з таким кроків:

1. Отримання кадру.
2. Виявлення облич методом Віоли-Джонса.
3. Застосування фільтра Гауса до знайдених облич.
4. Застосування маски значимих областей.
5. Розрахунок LBP гістограм знайдених облич.
6. Класифікація методом найближчого сусіда.
7. Візуальне виділення облич в кадрі.

З точки зору швидкодії ідеальним вибором в якості ознак класифікації при розробці системи розпізнавання в реальному часі являються центрально-симетричні локальні бінарні шаблони[5]. В результаті проведених досліджень LBP операторів з використанням бази зображень облич Кембриджського та Єльського університетів було зроблено висновок, що для досягнення точності $>90\%$ доцільно використовувати розбиття зображень від 4×4 і більше. Розбиття на 4×4 і більше - оптимальне рішення по співвідношенню точності і витрат пам'яті при розбитті зображення на області. Це розбиття забезпечує стабільно високий відсоток правильних класифікацій при не дуже великих витратах пам'яті.

Висновки

У статті запропоновано спосіб для розпізнавання облич у відеопотоках на основі метода Віоли-Джонса і локальних бінарних шаблонів. Для виявлення облич в кадрах відеопотоку був запропонований метод Віоли-Джонса. Класифікація виявлених облич виконувалась методом найближчого сусіда з використанням центрально-симетричних локальних бінарних шаблонів.

Тестування розробленої системи виявило результати в приблизно 94% правильних розпізнаних облич при обробці кадрів з відеопотоку з веб-камери реального часу.

Окрім цього було проведено дослідження ефективності різних модифікацій локальних бінарних шаблонів, які застосовувались для розпізнавання облич в реальному часі. Було встановлено, що центрально-

симетричні локальні бінарні шаблони можуть бути використані в якості ознак розпізнавання і при цьому мають більш високу швидкість роботи.

Також варто відмітити, запропонована архітектура системи розпізнавання облич дозволяє легко замінювати окремі модулі, що відкриває більше можливості для подальшого розвитку, модифікації та покращення системи.

Науковий керівник д.т.н, професор кафедри системного програмування і спеціалізованих комп'ютерних систем, Національний технічний університет України Київський політехнічний інститут», м. Київ, Україна, Зайцев В.Г.

Література

6. *H. A. Rowley, S. Baluja, T. Kanade.* Neural Network-Based Detection // PAMI, January 1998.
7. *P.Viola, M.Jones.* Rapid object detection using a boosted cascade of simple features / 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognitions Vol. 1. 8-14 December 2001 / The institute of Electrical and Electronics Engineers, Inc. C. 511-518.
8. *M. Heikkila, M. Pietikainen.* A texture-based for modeling the background and detecting moving objects // IEEE Transactions on Pattern Analysis and Machine Intelligence, 2006, №28(4), C. 657-662.
9. *M.Heihhila, M. Pietikainen. C. Schmid.* Description of Interest Regions with Center-Symmetric Local Binary Patterns // ICVGIP 2006, C.58-69.
10. *Ahonen, A. Hadid, M.Pietikainen.* Face Description with Local Binary Patterns: Application to Face Recognition // IEEE Trans. Pattern Analysis and Machine Intelligence, 1996, №28(12), C. 2037-2041.

ДОДАТОК Г. ФРАГМЕНТ ПРОГРАМНОГО КОДУ

```
//головний клас додатку
//підключення бібліотек
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.IO;
using System.Runtime.Serialization.Formatters.Binary;

//підключення бібліотеки EmguCV
using Emgu.CV;
using Emgu.CV.Structure; using Emgu.CV.UI;
using ZedGraph;

namespace CSLBPH
{
    public partial class MainForm : Form
    {
        private FaceDetector _detector; //детектор облич
        private List<FaceClass> _faces; //класи облич
        private List<RecognizedFace> _recognizedFaces; //розпізнані обличчя
        private List<Rectangle> _notRecognized; //нерозпізнані обличчя
        private Image<Gray, Byte> _mask; //маска значущих областей
        private Image<Gray, Byte>[] _currentFaces; //зображення для формування
        нового класу
        private int _currentFacesCount; //їх кількість
        private Rectangle[] _facesRect; //знайдені обличчя
        private bool _capturing; //флаг активності захвату відеопотоку
        private Capture _capture = null; //захват відеопотоку
        private String _videoPath = null; // шлях до відеофайлу
        private Mat _frame; //кадр відеопотоку
        private Image<Bgr, Byte> _frameImg; //кадр відеопотоку

        //конструктор класу public MainForm()
        {
            InitializeComponent();
        }
    }
}
```



```

_detector = new FaceDetector();
_faces = new List<FaceClass>();
_notRecognized = new List<Rectangle>();
_recognizedFaces = new List<RecognizedFace>();
_currentFaces = new Image<Gray, Byte>[1000];
_currentFacesCount = 0;
_facesRect = null;
_mask = new Image<Gray, byte>("mask.png");
_capturing = false;
_capture = new Capture();

zedGraphControl1.GraphPane.Title.Text = "Histogram";
zedGraphControl1.GraphPane.XAxis.Title.Text = "Value";
zedGraphControl1.GraphPane.YAxis.Title.Text = "Count";
}

//розрахунок гістограми
public Mat CalcHistogram(Image<Gray, Byte> image)
{
    int dim = 4; //розбиття на 4 блоки
    Mat result = new Mat();

    int width = image.Width / dim;
    int height = image.Height / dim;

    //ЦИКЛ по всім блокам
    for (int i = 0; i < dim; i++)
    {
        for (int j = 0; j < dim; j++)
        {
            image.ROI = new Rectangle(i * width, j * height, width, height);
            DenseHistogram hist = new DenseHistogram(16, new RangeF(0, 15));
            Image<Gray, Byte>[] inp = new Image<Gray, Byte>[1];
            inp[0] = image;

            //розрахунок для поточного блоку
            hist.Calculate(inp, true, null);
            CvInvoke.Normalize(hist, hist);

            //додавання до кінцевої гістограми
            result.PushBack(hist);
        }
    }

    image.ROI = Rectangle.Empty; return result;
}

```

```

}

private void RecognizeFace(Rectangle rect, Image<Gray, Byte> face,
Image<Gray, Byte> lbp, Mat histogram, decimal threshold)
{
double mindist = double.MaxValue
FaceClass answer = null;

foreach (FaceClass f in _faces)
{
foreach (Mat h in f.GetHist())
{
double dist = CvInvoke.CompareHist(histogram, h,
Emgu.CV.CvEnum.HistogramCompMethod.Chisqr);
if (dist <= mindist) {
mindist = dist
answer = f;
}
}
}

if (mindist < (double)threshold)
{
bool contains = false; CvInvoke.EqualizeHist(lbp, lbp);
foreach (RecognizedFace element in _recognizedFaces)
{
element.GetDist())

if ((element.GetFaceClass().GetName() == answer.GetName()) && (mindist <

{
_notRecognized.Add(element.GetRect());

element.SetFace(face); element.SetLBP(lbp); element.SetRect(rect);
element.SetHist(histogram); element.SetDist(mindist); contains = true;

}
else if ((element.GetFaceClass().GetName() == answer.GetName()) && (mindist
>= element.GetDist()))
{
contains = true;

_notRecognized.Add(rect);

```

```

    }
    }

    if (!contains)
    {
        mindist));

    }
    _recognizedFaces.Add(new RecognizedFace(answer, rect, face, lbp, histogram,

    }

    else
    {
        _notRecognized.Add(rect);
    }
    }

    void ProcessFaces()
    {
        foreach (Rectangle rect in _facesRect)
        {
            _frameImg.ROI = rect;
            Image<Gray, Byte> face = _frameImg.Convert<Gray, Byte>();

            _frameImg.ROI = Rectangle.Empty;

            face = face.Resize(128, 128, Emgu.CV.CvEnum.Inter.Linear);
            CvInvoke.GaussianBlur(face, face, new Size((int)blurUpDown.Value * 2 - 1,
            (int)blurUpDown.Value * 2 - 1), 0);

            Image<Gray, Byte> lbpNoMask = LBPTransformer.CSTransform(face);
            Image<Gray, Byte> lbp = new Image<Gray, byte>(128, 128, new Gray(0));
            CvInvoke.cvCopy(lbpNoMask, lbp, _mask);
            Mat histogram = CalcHistogram(lbp);
            RecognizeFace(rect, face, lbp, histogram, thresholdUpDown.Value);
            //распознавание
        }

    private void ProcessFrame(object sender, EventArgs arg)
    {
        _recognizedFaces = new List<RecognizedFace>();
        _notRecognized = new List<Rectangle>();
        _frameImg = new Image<Bgr, Byte>(_capture.Width, _capture.Height);
        _frame = new Mat();

```

```

_capture.Retrieve(_frame, 0);
_frameImg = _frame.ToImage<Bgr, Byte>();

_facesRect = _detector.GetFacesRect(_frameImg,
(double)scalefactorUpDown.Value, (int)neighborsUpDown.Value,
(int)minsizeUpDown.Value);

if (_facesRect.Length == 1)
{
addFaceButton.Invoke(new Action(() => { addFaceButton.Enabled = true; }));
}
else
{
addFaceButton.Invoke(new Action(() => { addFaceButton.Enabled = false; }));
}

ProcessFaces();
DrawDetected();}

private void ClearData()
{
zedGraphControl1.GraphPane.CurveList.Clear();
zedGraphControl1.AxisChange(); zedGraphControl1.Invalidate();
capFaceImageBox.Image = null; lbpImageBox.Image = null;

distLabel.Invoke(new Action(() =>
{
distLabel.Text = " ";
}));
}

private void DrawDetected()
{
bool contains = false;

foreach (RecognizedFace face in _recognizedFaces{
_frameImg.Draw(face.GetRect(), new Bgr(0, 255, 0), 3);
_frameImg.Draw(face.GetFaceClass().GetName(), face.GetRect().Location,
Emgu.CV.CvEnum.FontFace.HersheyComplex, 1, new Bgr(0, 255, 0));
faceClassesListBox.Invoke(new Action(() =>
{
if (face.GetFaceClass().GetName() == (string)faceClassesListBox.SelectedItem)
{

```

```

    }
    }));
    }
    capFaceImageBox.Image = face.GetFace(); lbpImageBox.Image = face.GetLBP();
    DrawHistogram(face.GetHist()); distLabel.Text = face.GetDist().ToString();
    contains = true;

    if (!contains) ClearData();

    foreach (Rectangle rect in _notRecognized) {
        _frameImg.Draw(rect, new Bgr(0, 0, 255), 3);
    }

    _frameImg = _frameImg.Resize(captureBox.Width, (int)(_frame.Height *
        (captureBox.Width / (double)_frame.Width)), Emgu.CV.CvEnum.Inter.Linear);
    captureBox.Image = _frameImg;
    }

    private void DrawHistogram(Mat histogram)
    {
        GraphPane pane = zedGraphControl1.GraphPane; pane.CurveList.Clear();
        int dim = histogram.Height;
        Image <Gray, Double> histimg = histogram.ToImage<Gray, Double>();

        double[] hist = new double[dim]; double[] xvalues = new double[dim];

        for (int i = 0; i < dim; i++)
        {
            xvalues[i] = i;
            hist[i] = histimg.Data[i, 0, 0];
        }

       BarItem bar1 = pane.AddBar("", xvalues, hist, Color.DeepSkyBlue);
        zedGraphControl1.AxisChange();
        zedGraphControl1.Invalidate();
    }

    private void openFaceClass(object sender, CancelEventArgs e)
    {
        foreach (string name in openFileDialog1.FileNames)
        {
            if (File.Exists(name))
            {
                Stream TestFileStream = File.OpenRead(name); BinaryFormatter deserializer =
                new BinaryFormatter();
            }
        }
    }

```

```

FaceClass face = (FaceClass)deserializer.Deserialize(TestFileStream);
TestFileStream.Close();
_faces.Add(face); faceClassesListBox.Items.Add(face.GetName());
}
}
}
void addFaceButton_Click(object sender, EventArgs e)
{
if ((_currentFacesCount < 1000) && (_facesRect.Length != 0))
{
for (int qwe = 0; qwe < 1000; qwe++)
{

Image<Gray, Byte> fr = _frame.ToImage<Gray, Byte>(); fr.ROI = _facesRect[0];

_currentFaces[_currentFacesCount] = fr.Resize(128, 128,
Emgu.CV.CvEnum.Inter.Linear);
CvInvoke.GaussianBlur(_currentFaces[_currentFacesCount],
_currentFaces[_currentFacesCount],
new Size((int)blurUpDown.Value * 2 - 1, (int)blurUpDown.Value * 2 - 1),
0);
imageBox4.Image = _currentFaces[_currentFacesCount];
_currentFacesCount++;
countLabel.Text = _currentFacesCount.ToString(); addClassButton.Enabled =
true; removeFaceButton.Enabled = true;
}
}
}
private void addClassButton_Click(object sender, EventArgs e)
{
Image<Gray, Byte>[] img = new Image<Gray, Byte>[_currentFacesCount];
Image<Gray, Byte> face = _currentFaces[0].Resize(128, 128,
Emgu.CV.CvEnum.Inter.Linear);
Mat[] hist = new Mat[_currentFacesCount];

for (int i = 0; i < _currentFacesCount; i++)
{
Image<Gray, Byte> lbpNoMask =
LBPTransformer.CSTransform(_currentFaces[i]); img[i] = new Image<Gray,
byte>(128, 128, new Gray(0)); CvInvoke.cvCopy(lbpNoMask, img[i], _mask);
hist[i] = CalcHistogram(img[i]);
}

_faces.Add(new FaceClass(face, hist, classNameTextBox.Text));

```

```

faceClassesListBox.Items.Add(classNameTextBox.Text);

_currentFaces = new Image<Gray, Byte>[1000];
_currentFacesCount = 0;
countLabel.Text = "0";

addClassButton.Enabled = false; removeFaceButton.Enabled = false;
imageBox4.Image = null;
}

private void faceClassesListBox_SelectedIndexChanged(object sender, EventArgs
e)
{
foreach (FaceClass face in _faces)
{
if (face.GetName() == (string)faceClassesListBox.SelectedItem)
{
classImageImageBox.Image = face.GetImg();
}
}

if (faceClassesListBox.SelectedItem != null)
{
saveClassButton.Enabled = true; removeClassButton.Enabled = true;
}
else
{
saveClassButton.Enabled = false; removeClassButton.Enabled = false;
}
}

private void removeFaceButton_Click(object sender, EventArgs e)
{
_currentFacesCount--;
countLabel.Text = _currentFacesCount.ToString(); if (_currentFacesCount != 0)
{
imageBox4.Image = _currentFaces[_currentFacesCount - 1];
} else
{
removeFaceButton.Enabled = false; addClassButton.Enabled = false;
imageBox4.Image = null;
}

private void videoButton_Click(object sender, EventArgs e)
{
if (_capturing)
{
startButton.Text = "Start";

```

```

    _capturing = false;
    _capture.Stop();
}
else
{
    try
    {

        startButton.Text = "Stop";
        _capturing = true;

        if (radioButton1.Checked)
        {
        }
        else
        {

            _capture.Dispose();
            _capture = new Capture();
            _capture.ImageGrabbed += ProcessFrame;
            _capture.Start();

            _capture.Dispose();
            _capture = new Capture(_videoPath);

            _capture.ImageGrabbed += ProcessFrame;
            _capture.Start();
        }
    }
    catch (NullReferenceException excpt)
    {
        MessageBox.Show(excpt.Message);
    }
}

}

private void saveFaceClass(object sender, CancelEventArgs e)
{
    FaceClass tosave = null;

    foreach (FaceClass face in _faces)
    {
        if (face.GetName() == faceClassesListBox.SelectedItem.ToString())
        {

```



```
tosave = face;
}
}
```

```
Stream fileStream = File.Create(saveFileDialog1.FileName); BinaryFormatter
serializer = new BinaryFormatter(); serializer.Serialize(fileStream, tosave);
fileStream.Close();
}
private void saveClassButton_Click(object sender, EventArgs e)
{
saveFileDialog1.ShowDialog();
}
```

```
private void openClassButton_Click(object sender, EventArgs e)
{
openFileDialog1.ShowDialog();
}
```

```
private void removeClassButton_Click(object sender, EventArgs e)
{
_faces.RemoveAll(f => f.GetName() ==
(string)faceClassesListBox.SelectedItem);
faceClassesListBox.Items.Remove(faceClassesListBox.SelectedItem);
}
```

```
private void openVideoButton_Click(object sender, EventArgs e)
{
openFileDialog2.ShowDialog();
}
```

```
private void openFileDialog2_FileOk(object sender, CancelEventArgs e)
{
_videoPath = openFileDialog2.FileName; videoPathLabel.Text = _videoPath;
}

}
}
```